HUMAN-ROBOT INTERACTION AND COMPUTER-VISION-BASED SERVICES FOR AUTONOMOUS ROBOTS

Jordi Bautista Ballester

HUMAN-ROBOT INTERACTION AND COMPUTER-VISION-BASED SERVICES FOR AUTONOMOUS ROBOTS

DOCTORAL THESIS

Author:
Jordi Bautista i Ballester

Advisors:
Dr. Domènec Savi Puig Valls
Dr. Jaume Vergés Llahí

Department of Computer Engineering and Mathematics

Intelligent Robotics and Computer Vision Group (IRCV)

UNIVERSITAT ROVIRA I VIRGILI
Tarragona
2016

"Supported by Pla de Doctorats Industrials de la Secretaria d'Universitats i Recerca del Departament d'Economia i Coneixement de la Generalitat de Catalunya"

UNIVERSITAT ROVIRA I VIRGILI
Departament d'Enginyeria Informàtica i Matemàtiques
Av. Paisos Catalans, 27
43007 Tarragona
Tel. +34 977 55 95 95
Fax. +34 977 55 95 97

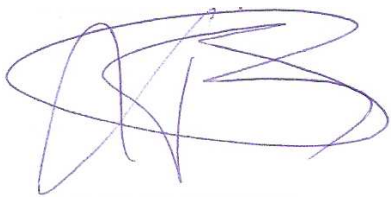
FAIG CONSTAR que aquest treball, titulat "INTERACCIÓ ROBOT-HUMÀ I SERVEIS DE VISIÓ PER COMPUTADOR PER A ROBOTS AUTÒNOMS", que presenta en Jordi Bautista i Ballester per a la obtenció del títol de Doctor, ha estat realitzat sota la meva direcció al Departament d'Enginyeria Informàtica i Matemàtiques d'aquesta universitat i que acompleix els requeriments per poder optar a Menció Internacional.

HAGO CONSTAR que el presente trabajo, titulado "INTERACCIÓN ROBOT-HUMANO Y SERVICIOS DE VISIÓN POR COMPUTADOR PARA ROBOTS AUTÓNOMOS", que presenta Jordi Bautista i Ballester para la obtención del título de Doctor, ha sido realizado bajo mi dirección en el Departamento de Ingeniería Informática i Matemáticas de esta universidad y que cumple con los requerimientos para poder optar a Mención Internacional.

I STATE that the present study, entitled "HUMAN-ROBOT INTERACTION AND COMPUTER-VISION-BASED SERVICES FOR AUTONOMOUS ROBOTS", presented by Jordi Bautista i Ballester for the award of the degree of Doctor, has been carried out under my supervision at the Department of Computer Engineering and Mathematics of this university and that fulfills the requirements to opt the International Mention.

Tarragona, 15th April 2016.

El/s director/s de la tesi doctoral
El/los director/es de la tesis doctoral
Doctoral Thesis Supervisor/s



Dr. Domènec Savi Puig Valls

Dr. Jaume Vergés Llahí

UNIVERSITAT ROVIRA I VIRGILI
HUMAN-ROBOT INTERACTION AND COMPUTER-VISION-BASED SERVICES FOR AUTONOMOUS ROBOTS
Jordi Bautista Ballester

Software Department

C. Víctor Pradera, 45

08940 Cornellà de Llobregat

Tel. +34 93 204 99 22

Fax. +34 93 204 98 66

I STATE that the present study, entitled "HUMAN-ROBOT INTERACTION AND COMPUTER-VISION-BASED SERVICES FOR AUTONOMOUS ROBOTS", presented by Jordi Bautista i Ballester for the award of the degree of Doctor, has been carried out in conjunction with the "VINBOT" project developed at the company, under my supervision at the Department of Software, Robotics and Computer Vision of this company.

His role as Research Engineer in FP7 Vinbot project demanded that he works harmoniously with a wide range of people while commanding respect. He had a positive attitude when dealing with both his superiors and his colleagues. He always displayed a responsible attitude toward his tasks either in Vinbot project and other tasks commanded; he demanded a great deal of himself and he worked hard to ensure professional task performance. After spending three years in Ateknea, he has proven himself as a valuable member of our company with great ideas and strong interpersonal skills.

In particular, concerning VINBOT project, I confirm the positive feedback received from the Research Executive Agency with its Assessment Report delivered in December, 2015. In this report is stated that a good progress is achieved in the first reporting period of the project. Hence the 90% of the funding are delivered by the European Commission for the project, which is an uncountable merit of the whole VINBOT team. In particular, concerning Jordi's work specific achievements are highlighted, stating that major aspects of the technical solution have been performed accordingly, both concerning the sensor head developed in our company by Jordi and the proposed robotic platform. The report also states that ongoing development seems in line with the goals of the project and the objectives for the period have been achieved with a satisfactory progress in relation to the description of work. Furthermore, the results of the vine developmental parameters are convincing and the preliminary results on automatic measurements are encouraging. The dissemination was performed in several fairs and industrial exhibitions.

Cornellà de Llobregat, 15 April 2016

Company's supervisor/s,

Albert Rodríguez Pastor

UNIVERSITAT ROVIRA I VIRGILI
HUMAN-ROBOT INTERACTION AND COMPUTER-VISION-BASED SERVICES FOR AUTONOMOUS ROBOTS
Jordi Bautista Ballester

To my parents, Joan and Mª Teresa
To my brother Carles and my sister Cristina
To my niece Laura, the most brighten light in my heart
To my dear Núria

UNIVERSITAT ROVIRA I VIRGILI
HUMAN-ROBOT INTERACTION AND COMPUTER-VISION-BASED SERVICES FOR AUTONOMOUS ROBOTS
Jordi Bautista Ballester

Abstract

Imitation Learning (IL), or robot Programming by Demonstration (PbD), covers methods by which a robot learns new skills through human guidance and imitation. PbD takes its inspiration from the way humans learn new skills by imitation in order to develop methods by which new tasks can be transmitted to robots. This thesis is motivated by the generic question of "*what to imitate?*" which concerns the problem of how to extract the essential features of a task. To this end, here we adopt Action Recognition (AR) perspective in order to allow the robot to decide what has to be imitated or inferred when interacting with a human kind.

The proposed approach is based on a well-known method from natural language processing: namely, Bag of Words (BoW). This method is applied to large databases in order to obtain a trained model. Although BoW is a machine learning technique that is used in various fields of research, in action classification for robot learning it is far from accurate. Moreover, it focuses on the classification of objects and gestures rather than actions. Thus, in this thesis we show that the method is suitable in action classification scenarios for merging information from different sources or different trials.

This thesis makes three contributions: (1) it proposes a general method for dealing with action recognition and thus to contribute to imitation learning; (2) the methodology can be applied to large databases which include different modes of action captures; and (3) the method is applied specifically in a real international innovation project called VinBot.

Keywords: Imitation Learning, Sensor Fusion, Robotics, Action Recognition, Human Robot Interaction, Computer Vision, Bag of Words, Multikernel SVM.

Acknowledgements

I would like to express my gratitude to Dr. Domenec Puig, from the Intelligent Robotics and Computer Vision Group (IRCV), for his continuous support and advice during my PhD, for guiding my research and offering me the opportunities to present my work at conferences and workshops. The acknowledgement would not be complete without thanking him for the time that he devoted for managing this Industrial Doctorate Program.

I would also like to thank Dr. Jaume Vergs for his co-supervision at Ateknea Solutions Catalonia, previously known as Centre de Recerca i Innovació de Catalunya (CRIC). His advice and supervision at the company were a great opportunity for me to improve on my previous professional experience and develop my scientific skills in industrial environments. Furthermore, I thank Ateknea Solutions Catalonia for trusting in me to be a member of their team for the Industrial Doctorate Pilot Plan as well as my supervisor Albert Rodríguez for making this thesis feasible.

I must not forget to give a very special acknowledgement to Prof. Aude Billard, who gave me the opportunity to work in a great scientific environment and enjoyable atmosphere at the Learning Algorithms and Systems Laboratory (LASA) during my stay at the École Polythecnique Fédéral de Lausanne (EPFL), in Switzerland.

I also acknowledge all of my collaborators, my mates at the IRCV, the Vinbot team in Ateknea and the LASA members, for the support and friendship that they offered me during these three years. My deepest gratitude to Julian and to Pablo, whose support at the very beginning of this PhD gave me the strength to carry on with it. My thanks also go to my parents, Joan i M^{a}Teresa, my brother Carles and my sister Cristina, for their invaluable support; my niece, Laura, who give me the light in this life with her joy and love; and all my family and friends for their affection. And finally, I must make special mention of my lovely Núria, the unconditional love and comprehension of whom have brighten me the path to this end.

Thanks also to the reviewers of conference papers and my thesis, all of whose advice has helped improve the quality of my work.

This research has been partially funded by the Industrial Doctorate programme of the Government of Catalonia, and by the European Community which funded the Vinbot project (Nº605630) conducted by Ateknea Solutions Catalonia through the FP7 framework programme. The École Polythecnique Fédéral de Lausanne also contributed through the Robohow project (Nº288533) by funding my stay in Switzerland.

Contents

Acronyms

2D	2 Dimensions. 19, 92
3D	3 Dimensions. 3, 4, 18, 19, 28, 55–57, 59–61, 63–65, 68, 69, 71, 72, 75–77, 83–85, 89, 90, 92, 93, 95, 102, 103, 112, 113, 119, 123–128, 130, 133, 137–142, 144–147
API	Application Program Interface. 130, 131
AR	Action Recognition. 6, 7
ARBoW	Action Recognition based on BoW. 152, 157, 158
BM	Binary Merging. 14
BoVW	Bag of Visual Words. 25, 31
BoW	Bag of Words. vii, 2, 7, 25, 26, 29, 32, 33, 36, 39–43, 46, 53, 55, 56, 59, 63, 64, 67, 71, 76, 79–84, 90, 101, 102, 116, 135, 136, 157
BP-HMM	Beta Process Hidden Markov Model. 153, 158
BTA	Barcelona Tecnologias de la Alimentacion. 146, 147
CAD120	Caltech Action Database. 3, 56, 57, 60, 63, 65–72, 75, 76, 85, 88–90, 92–95, 97, 98, 103, 107, 108, 112, 113
CB-SVM	Cluster Based Support Vector Machine. 79
CMMKL	Contextual and Modal MultiKernel Learning. 68, 70–72, 93

CMMKL-SVM	Contextual and Modal MultiKernel Learning Support Vector Machine. 55, 59, 64, 66, 67, 69, 71, 72, 91, 92, 102
CPU	Central Processing Unit. 20, 130
CRF	Conditional Random Fields. 69, 71
DC	Direct Current. 128
DMP	Dynamic Movement Primitives. 14
DOF	Degrees of Freedom. 125, 128
EM	Expectation Maximization. 13
EPFL	Ecole Politechnique Federale de Lausanne. 119
EPFL-LfDD	EPFL Learning from Demonstration Database. 154, 156
EU	European Union. 120, 121, 146
FAST	Features from Accelerated Segment Test. 34
FP7	7th Framework Program. 4
FPFH	Fast Point Feature Histograms. 60, 61, 64, 67–70, 72, 83–85, 90–94, 98, 102, 113
GIST	Global Image Spectrum Templates. 45, 58, 61, 85
GMM	Gaussian Mixture Model. 13, 14
GMR	Gaussian Mixture Regression. 13, 14
GPR	Gaussian Process Regression. 13, 14
GPS	Global Positioning System. 123, 125–127, 130, 132, 143, 144

GPU Graphical Processing Unit. 19, 20

HC-SVM Convex Hull Support Vector Machine. 78

HDD Hard Disk Drive. 155, 156, 159

HMDB Human Motion Database. 3, 39–41, 47, 48, 50–52, 56, 57, 63, 65–72, 75, 77, 85, 88–90, 92–95, 97, 98, 103, 107, 108, 110, 111

HMM Hidden Markov Models. 14, 19

HOF Histogram of Optical Flow. 19, 26, 42, 45–47, 49–51, 53, 58, 60, 61, 63, 64, 67–70, 72, 83–85, 90–93

HOG Histogram of Gradients. 18, 19, 26, 27, 42, 45–47, 49–51, 53, 58, 60, 63, 64, 67–72, 83–85, 90–94, 98, 102, 157–159

HOG3D 3 Dimensional Histogram of Gradients. vii–ix, 27, 29, 34–37, 41, 42, 45–47, 49–52, 58, 61, 63, 64, 67, 68, 83–85, 90–92

HOHA Hollywood Human Actions. 20

IL Imitation Learning. 15

IMU Inertial Motion Unit. 108, 125, 126, 128, 130, 132, 143

IP Interest Point. vii, viii, 26, 29, 33, 34, 37

IPs Interest Points. vii, ix, 26, 27, 29, 34, 135

IRL Inverse Reinforcement Learning. 15

ISA Instituto superior de Agronomia. 122, 146, 147

ISVM Incremental Support Vector Machine. 78

IWCMMKL	Incremental Weighted Contextual and Modal MultiKerneL. 94
IWCMMKL-SVM	Incremental Weighted Contextual and Modal MultiKerneL Support Vector Machine. 75, 94, 95, 117
KLT	Kanade Lucas Tomasi. 45, 83
kNN	k-Nearest Neighbor. 26, 44, 82, 154, 156
KTH	Kungliga Tekniska Hogskolan. 3, 19, 34, 39, 40, 46–48, 50–52, 63, 64, 90, 103, 107–109
LAI	Leaf Area Index. 137, 139–141, 145
LASA	Learning Algorithms and Systems Laboratory. 119, 151
LASVM	Learning Active Support Vector Machine. 79
LfD	Learning from Demonstration. 103, 149
LP	Learning Prototypes. 78, 79
LRF	Laser Range Finder. 125, 143
LSV	Learning Support Vectors. 79
LSVM	Lagrangian Support Vector Machine. 79
LWPR	Locally Weighted Projection Regression. 13, 14
MBH	Motion Boundary Histogram. 42, 45–47, 49–51, 53, 60, 61, 63, 64, 67–70, 72, 83–85, 90–93
MIL	Multiple Instance Learning. 42, 58
MKL	Multiple Kernel Learning. 59, 69, 72, 92, 105

MMM Master Motion Map. 108

MoCap Motion Capture. 150, 152, 155

MoMP Mixture of Motor Primitives. 14

NDVI Normalized Difference Vegetation Index. 123–127, 133, 134, 137, 138, 144

NIC Network Interface Controller. 129, 130

NIR Near InfraRed. 123, 125, 126, 132, 133, 137, 144

NL Natural Language. 108

NMEA National Marine Electronics Association. 125

NORMA Naive Online Rreg Minimization Algorithm. 78

OF Optical Flow. 45, 83

OI-SVM Online Incremental Support Vector Machine. 78, 79

ORBoW Object Recognition based on BoW. 134–136, 144, 145

OS Operating System. 130

PbD Programming by Demonstration. 1, 2, 6, 7, 9, 11, 13, 99–101, 104, 105, 115

PC Personal Computer. 127–129, 143, 156

PCL Point Cloud Library. 140, 144

PDA Personal Digital Agenda. 127

PPK Probability Product Kernel. 80

PV Precision Viticulture. 120

QP Quadratic Programming. 77

RANSAC Random Sample Consensus. 31, 44, 82
RBF Radial Basis Functions. 36, 37, 59, 79
RFID Radio Frequency Identification. 108
RGB Red-Green-Blue. 2, 4, 55–57, 60, 61, 63–65, 67–71, 75–77, 81, 83–85, 89, 90, 92, 93, 95, 102, 103, 108, 113, 125, 126, 132, 133, 137, 138, 141, 144, 145, 152, 154, 155, 157
RGB-D Red-Green-Blue-Depth. 60, 63, 65, 84, 89, 90, 108, 112, 113, 150, 152, 153, 155
RL Reinforcement Learning. 11, 12, 15
RMSE Root Mean Square Error. 12, 14
RMST Relative Multi Scale Tracklet. 42
ROI Region of Interest. 17
ROS Robot Operating System. 131, 154, 156
RTD Research, Technological Development and Innovation. 121, 122

SA Sectional Area. 140–142
SCM Super Cluster Machine. 80
SDK Software Development kit. 130
SEDS Stable Estimator of Dynamical Systems. 14
SIFT Scale Invariant Feature Transform. 19, 45, 83, 135, 136
SME Small and medium-sized enterprise. 121, 122
SME-AGs SME Associations. 121, 122, 145

SMST Shape Multi Scale Tracklet. 42

SSD Solid State Disk. 159

STIP Spatio Temporal Interest Points. 26, 44, 46, 64, 81, 83, 90

SURF Speeded Up Robust Features. 34, 46, 63, 82, 89, 135, 136

SV Support Vectors. 62, 75, 77–80, 87, 102

SVM Support Vector Machine. vii–ix, 26, 29, 32, 34–37, 39, 41–45, 47, 49, 53, 55, 58, 59, 61, 62, 64, 67, 70, 75–82, 84, 86–88, 91, 102, 117, 136, 157

TAI Tree Area Index. 141

UAV Unmanned Aerial Vehicles. 17, 128

UCF University of Central Florida. 107

UFC University of Central Florida. 20

USB Universal Serial Bus. 129

UW Uniformly Weighted. 68, 70–72

WP Work Package. 124

WPs Work Packages. 124

List of Figures

List of Tables

Chapter 1

Introduction

"Begin at the beginning, and go on till you come to the end: then stop."

- Lewis Carroll, *Alice in Wonderland*

Since the 1980s research into Programming by Demonstration (PbD) has grown steadily and become a central topic in robotics. Complex platforms that interact in complex and variable environments are faced with two key challenges when learning robot skills.

First, the complexity of the task is such that learning only by trial-and-error would be impractical. Therefore, PbD is a strategy that can speed-up and facilitate the process of learning by reducing the search space and allowing the robot to refine its model of demonstration by trial-and-error. PbD also permits the robot to incorporate usual tasks by means of a non-specialized instructor.

Second, PbD favors a closer relation between the learning process and the control stage, so the latter can be adapted in real time to perturbations and changes that are likely to happen in the environment.

PbD covers methods by which a robot learns new skills through human guidance and imitation. Also referred to as *imitation learning, lead through teaching, tutelage, or apprenticeship learning*, PbD takes inspiration from the way humans learn new skills by imitation in order to develop methods by which new tasks can be transmitted

to robots. In this paradigm, the programmer becomes an instructor who is either a human being or another robot. The skill is then decomposed and programmed by observation of a demonstration performed by this instructor .

The challenges faced by PbD were enumerated in Nehaniv and Dautenhahn (2001) as a set of key questions: *What to imitate? How to imitate? When to imitate? Whom to imitate to?* To date only the first two questions have actually been addressed in PbD.

In a similar manner, Breazeal and Scassellati (2002) went further into how, in fact, a robot can know what elements to imitate when attempting to learn the movements of a human being, what perceptual aspects are relevant to this task and how a robot can distinguish the parts that will be emulated once a particular action is perceived, as well as the moment the robot must convert this perception into a sequence of motor responses in order to achieve the same result.

According to the work by Argall et al. (2009), PbD can be divided in two stages, namely, the collection of examples, which contain all the information that make up a demonstration, and the derivation of a policy, also known as mapping, by which a set of examples is used to define a group of actions that can reproduce the behavior outlined in the demonstration.

Hence, video analysis has become critical in human robot interactions, from which a robot must make a decision by considering the information extracted from robot joints sensors, accelerometers, cameras or lasers. In this context, our research focuses recognizing action in videos containing multimodal and contextual information. Furthermore, we extend the traditional Bag of Words (BoW) approach: first, to a descriptor encoder, adding contextual information extracted from RGB sequences, second, to a fusion of multiple information sources, incorporating information coming from non visual sensors, and third, to an incremental learning approach, so that new data can be incorporated to the trained model. The approach is applicable to different fields of research: for example, imitation learning, where the approach allows the robot to learn even from its own performance of an imitation, or in such other real applications as action segmentation.

In order to test this new approach we focused on experimentation and made use of well known recorded and collected databases. Public databases are usually made up of a set of videos where scenes and parameters such as illumination, focus, distance, and viewpoints are mostly controlled, and there is little information about the tools and objects that are involved in the action. For example, the Kungliga Tekniska Hogskolan (KTH) database created by Schuldt et al. (2004), a popular choice for testing action recognition techniques, contains this kind of information but not much about of color, context and modes. Furthermore, as the need for more sophisticated data increases in some fields of research (e.g. in robotic environments) multimodal information, provided by distance laser sensors or by 3D cameras such as Kinect, has been incorporated into more recent databases.

One of the most complete database is the Caltech Action Database (CAD120) created by Koppula et al. (2013). It is recorded in a high controlled environment, which is ideal for human-robot interactions and it includes both contextual and multimodal information. The database contains 10 high level actions performed by 4 different subjects which in total make up 124 manually annotated videos.

However, in order to go beyond the current state of the art in action recognition for real videos, more realistic databases have increasingly been employed, including videos that stage more realistic actions. The Human Motion Database (HMDB) by Kuehne et al. (2011), is one of the largest action video database to-date with 51 action categories, which in total contains 6766 manually annotated clips extracted from a variety of sources ranging from digitized movies to YouTube. This database has been created to evaluate the performance of computer vision systems for action recognition and explore the robustness of these methods under various conditions such as cluttered backgrounds, fast irregular motions, occlusions and camera motion. Although no multimodal recording is available, actions with contextually connected objects can be found in this database.

From the Human Motion Database (HMDB), we selected a subset of actions that are performed using a tool or object. This contextual information allows the computer to discriminate apparently similar actions such as shooting a gun or a bow.

The biggest difference between these similar actions lies in the tool used to carry out the action.

In this thesis, the different sources of information in the databases -depth, 3D and objects- are combined in a richer description of human actions that permits higher recognition rates. In order to increase the robustness of the recognition of actions in more challenging situations, we weight different sources of information that can be used to discriminate actions: namely, the spatio-temporal features that describe motion by RGB, depth and 3D modes, and the contextual information that explains how an action is carried out by object features. Finally, one of the most interesting contributions of the work is the proposal of an incremental approach that makes it possible to add new training data to the classifier without having to train it again from batch, something that is usually required when learning from demonstration.

1.1 Motivation

This thesis was motivated by our current research on envisaging the most relevant techniques that allow us to teach and share skills with a group of autonomous outdoor mobile robots called **VinBot**[2].

The purpose of the project is to provide the necessary navigation and behavioral skills to a number of agriculture robots so they can perform tasks such as monitoring the growth of crops and estimating such valuable information as yield. These robots will be networked: that is, they will be connected to a cloud-based service which will provide the off-board computational resources and all the necessary tools for communicating, storing, processing and sharing the data obtained by the on-board sensors.

VinBot robots must be capable of learning certain skills involved in autonomous outdoor navigation, such as creating maps of fields and coping with changes in this mutable environment while moving through them in different periods of time. Also

[2]VinBot is an European project from the 7th Framework Program (FP7) starting in 2014 whose main object is to develop a cloud-based mobile robotic system for agricultural applications. This project is explained in detail in Appendix C.

avoiding unknown obstacles and potential risks to the integrity of the robot will require learning new skills or, at least, sharing sets of strategies already acquired from other sources. Finally, we need natural ways to specify and control the missions, and to be able to teach certain tasks so that they can tend a specific crop.

Here the importance of autonomously learning skills from demonstrations is even more evident since interaction will mostly be with users who are unaware of robot programming and who might not even be in the same place as the robot. Our aim is also for this newly acquired knowledge to benefit other robots in the system. This implies that the representation of skills must be such that it can be transferred not only to similar robots but also to those that are not strictly identical, and it must also be shared and replicated among a number of units.

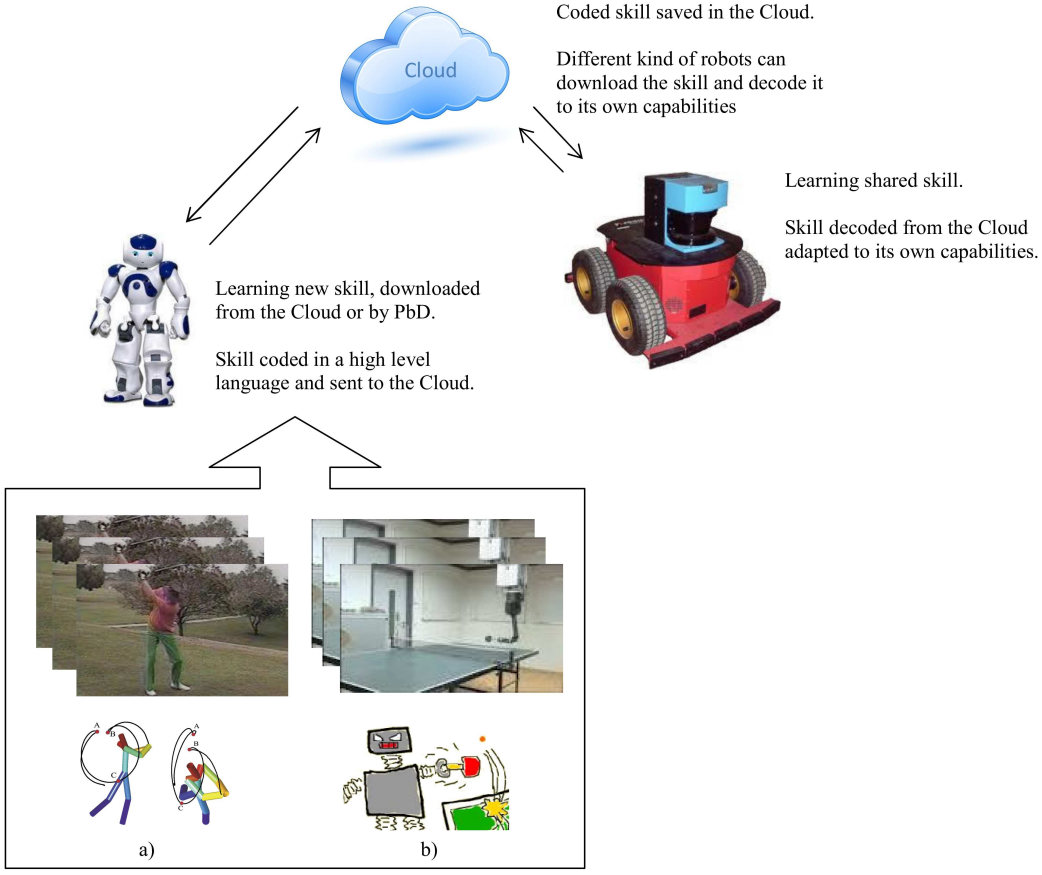


Figure 1.1: Robot understanding of reality. Learning a new task can be done by PbD: (a) from a video database of demonstrations, extracting macro actions or (b) from a set of demonstrations, composing basic actions in order to find an appropriate policy for the reproduction of the task. Afterwards, the learned task can be shared through the cloud to other robots with the same or different embodiment.

Consequently, we intend to use Programming by Demonstration to learn and transfer skills in the context of networked autonomous mobile robots. Thus, PbD is the natural approach to the problems of learning skills from demonstrators and representing skills among different robotic embodiments. The Figure 1.1 illustrates the problem statement that motivates this thesis. Although most of the approaches analyzed are usually applied to more human-like platforms, such as humanoids or robotic arms, we also investigate what type of approaches best fit our specific mobile robot platform.

1.2 Objectives

Bearing in mind the above discussion about PbD methods applied in robotics and the Action Recognition (AR) in computer vision, the goals of this thesis are:

- To assess the state of the art in PbD and understand which techniques are used to identify and extract the information required in order to make an inference or imitation. Furthermore, as we focus on the "what to imitate?" part of PbD, we move on to action recognition computer vision topic in order to respond to the more recent and challenging unsolved problems.
- To develop and validate an action recognition engine to solve the "what to imitate?" question in PbD by proposing a method of imitation learning based on instructional videos, in which the robot can learn a new skill from a set of videos. To do so, a machine learning approach will be used to extract actions from the instructional videos.
- To adapt the engine developed in real applications. The aim is to take advantage of the Industrial nature of this thesis to carry out experimentation in real applications. Furthermore, the theoretical approach developed and validated in this dissertation is adapted to a complex industrial project with promising results. However, its industrial property protection means that the results are not provided here: a mere overview of the experiments is given.

1.3 Overview

The rest of this thesis is organized as follows:

Chapter 2 introduces some fundamental concepts and notation and reviews the relevant literature on both PbD and AR.

Chapter 3 presents the traditional BoW approach, analyzing its three specific phases with particular emphasis on the parameter variation in each of the following phases: interest points detection and descriptor extraction; codebook generation; pooling and classification.

Chapter 4 presents and validates the introduction of context information relevant to the action into the BoW-based representation of action. Specifically, the context used in the chapter concerns the objects that are related to the action performed.

Chapter 5 presents and validates how the best information related to the action performed can be autonomously fused and selected by means of a BoW-based representation of the action.

Chapter 6 presents and validates an incremental approach that allows new training data to be incorporated into the classifier without having to train it again from batch.

Finally, Chapter 7 summarizes the contributions of the thesis and proposes future research directions and applications of the new concepts introduced in this thesis.

Additionally, we enclose four appendixes which are organized as follows:

Appendix A presents the databases used by the scientific community to experiment for action recognition purposes and specifically, it details the databases used in this dissertation.

Appendix B summarizes the contributions done to the scientific community in the form of book chapters, conference proceedings and journal papers.

Appendix C presents the industrial projects in which our scientific contributions have been adapted for real applications. In particular, we have been working in VinBot project for two and a half years as well as in RoboHow project for three months.

Appendix D summarizes the experience obtained by having performed this thesis in its industrial format.

Chapter 2

Fundamentals

"Those who do not want to imitate anything, produce nothing."

- Salvador Dalí,

2.1 Imitation Learning

The challenges faced by PbD were enumerated in Nehaniv and Dautenhahn (2001) as a set of key questions: *What to imitate? How to imitate? When to imitate? Whom to imitate to?* To date, only the first two questions have actually been addressed in PbD.

2.1.1 What to Imitate: Collection of Examples.

In this first stage, a set of information from the demonstrator, be it a robot or human, and possibly also from the environment, is collected from the readings of a capturing system. This can be a device mounted either on the demonstrator or on the learner, the commands of a remote control operated by the demonstrator, or a sensor located externally in the environment, like a camera.

Due to the correspondence problem – i.e., how to correspond actions in different embodiments and robotic platforms (Nehaniv and Dautenhahn, 2001) – in the collection stage we must be aware of the particular structure for both the demonstrator and the robot learner. Consequently, two successive mapping steps

Table 2.1: What to imitate? Techniques in the first stage –Collection of Examples according to Argall et al. (2009)–. The second column gives the method the demonstrator uses to convey the information to the learner. The third and fourth columns show whether record or embodiment mappings were used. The most extreme cases are teleoperation, which requires no mapping, and external observation, which requires both mappings.

Demonstrator Methods	Technique Names	Record Mapping	Embodiment Mapping	Related Works
Robot Learner	Teleoperation	No	No	Pook and Ballard (1993); Abbeel and Ng (2004); Chernova and Veloso (2008b); Breazeal et al. (2006); Rybski et al. (2007); Argall et al. (2007); Grollman and Jenkins (2007)
	Shadowing	Yes	No	Nehmzow et al. (2007); Ogino et al. (2006)
Externally	Sensors on Instructor	No	Yes	Ijspeert et al. (2002b); Calinon and Billard (2007); Aleotti and Caselli (2006)
	External Observation	Yes	Yes	Billard and Matarić (2001); Ude et al. (2004); Steil et al. (2004); Ratliff et al. (2006)

are required: namely, record mapping and embodiment mapping. The former maps sensor readings onto motor commands and the latter, maps motor commands from the demonstrator's body onto those of the learner.

Below we describe the techniques summarized in Table 2.1 showing their requirements with respect of record and embodiment mappings. There are two main groups of techniques which are differentiated by whether the demonstrator conveys directly the learner.

In teleoperation, the demonstrator operates the robot learner and its sensors capture the motion. No record mapping is necessary since the sensory system is the robot learner's. In shadowing, the demonstrator carries out the task and the robot learner captures the motion with its sensors and attempt to repeat it. In this case, a record mapping is necessary. Both in teleoperation and in shadowing no embodiment mapping is needed since the robot captures information directly from its sensors.

If the instructor is wearing sensors, the recording comes directly from these sensors and there is no need for any record mapping. With this imitation technique,

Table 2.2: How to imitate? Methods in the second stage –Deriving a Policy according to Argall et al. (2009)–. Most of the work done on PbD are in the Mapping Function category, and focus on either Classification or Regression.

Approaches	Learning Techniques		Related Works
Mapping Function	Classification	Low Level Robot Actions	Chernova and Veloso (2007); Saunders et al. (2006); Khansari-Zadeh and Billard (2011)
		Robot Movement Primitives	Pook and Ballard (1993); Billard and Matarić (2001); Kober and Peters (2009); Calinon et al. (2010b)
		High Level Behaviors	Chernova and Veloso (2007); Rybski and Voyles (1999); Lockerd and Breazeal (2004)
	Regression (Mapping Function Approx.)	At Run Time	Argall et al. (2007); Ijspeert et al. (2002a)
		Prior Run Time	Calinon et al. (2010b); Vijayakumar and Schaal (2000); Grollman and Jenkins (2008)
		Prior Execution Time	Grollman and Jenkins (2007); Calinon and Billard (2007); Ude et al. (2004); Steil et al. (2004)
System Models	Reward Based Learning	Engineering Reward Function	Chersi (2012); Merrick (2012)
		Learned Reward Function	Abbeel and Ng (2004); Ratliff et al. (2006); Guenter and Billard (2007); Abbeel et al. (2007)
Plans	Using a Planner		Rybski et al. (2007); Nicolescu and Mataric (2003)

the demonstrator executes the task and the sensory system information is recorded. If sensor readings come from an external observation, the demonstrator executes the task and the external sensory system records the execution which will be translated by means of a record mapping onto the learner motor commands. If the robot learner has no sensory device, an embodiment mapping is also required in both sensor-on-instructor and external-observation techniques.

2.1.2 How to Imitate: Derivation of a Policy.

The second stage consists of executing a group of actions that allows the robot to reproduce the behavior that was demonstrated by a set of examples. The following are the three most common approaches: 1) learning a function by mapping states to actions; 2) learning a model of world dynamics; or 3) using a planner that produces the sequence of actions after learning the model of an action.

We list the most interesting studies on these issues in Table 2.2, and classify them according to the learning approach used.

In the first approach, learning a mapping function, the algorithms can be grouped in two different families depending on whether the output is discrete (classification) or continuous (regression).

In the second approach, learning a model of world dynamics, a reward function is maximized. It can be user-defined or learned in an optimization process. It is typically formulated within the framework of Reinforcement Learning (RL).

In the third approach, goal executions can be represented as plans. Therefore, the planning framework represents a policy as a sequence of actions leading from an initial state to the target state.

2.1.3 Performance of Programming by Demonstration

Recent studies demonstrate that the performance of the techniques discussed can vary substantially. Different metrics can be used to compute the imitation performance for each of the approaches proposed. for example, Calinon et al. (2010a) used several metrics to evaluate an attempt reproducing the set of demonstrations (e.g. Root Mean Square Error (RMSE), Norm of Jerk, Learning Time and Retrieval Duration).

Argall et al. (2009) discusses the limitations of the dataset provided in the demonstration. The first limitation is caused by dataset sparsity, which can occur when the demonstrator cannot demonstrate all possible states (underdemonstrated states). To deal with this problem, Smart (2002) and Nicolescu and Mataric (2003) proposed a generalization from existing demonstrations and Chernova and Veloso (2008b), Grollman and Jenkins (2007) and Chernova and Veloso (2007), proposed an acquisition of new demonstrations. The second limitation is caused by the poor quality of a set of examples, which can happen whenever the instructor's demonstrations are ambiguous, unsuccessful or suboptimal in certain areas of the state space.

Various solutions have been proposed for improving demonstration data when

Table 2.3: Evaluation of the Apprenticeship. Two main reasons were identified in Argall et al. (2009) as the cause of low performance: underdemonstrated states and poor quality data. We show some of the works dealing with this problems and the approaches followed.

Reasons	Approaches	Related Works
Underdemonstrated State	Generalization from Existing Demonstration	Smart (2002); Nicolescu and Mataric (2003)
	Acquisition of New Demonstrations	Chernova and Veloso (2008b); Grollman and Jenkins (2007); Chernova and Veloso (2007)
Poor quality data	Suboptimal or Ambiguous Demonstrations	Pook and Ballard (1993); Breazeal et al. (2006); Aleotti and Caselli (2006); Chernova and Veloso (2007, 2008a)
	Learning from Experience	Smart (2002); Argall et al. (2007); Calinon and Billard (2007); Nicolescu and Mataric (2003); Chernova and Veloso (2008a)

the demonstration is suboptimal or ambiguous (Pook and Ballard (1993), Breazeal et al. (2006), Aleotti and Caselli (2006), Chernova and Veloso (2007) and Chernova and Veloso (2008a)) and attempts have been made to learn from experience by means of feedback from the demonstrator or a reward function, likewise it is done in Reinforcement Learning approaches (Smart (2002); Argall et al. (2007); Calinon and Billard (2007); Nicolescu and Mataric (2003); Chernova and Veloso (2008a)). Table 2.3 shows the main studies on the causes of low performance.

2.1.4 Taxonomy of Programming by Demonstration

Many approaches have been proposed for PbD. In this section, we discuss the most important ones (see Table 2.4) and we describe their advantages and drawbacks.

A vast majority of the approaches in PbD focus largely on the spatial position and velocity of the end effector or the joint angles. The first attempts depended on an explicit temporal indexing and virtually operated in an open loop. The main drawback of these techniques is that time dependence makes them very sensitive to both temporal and spatial perturbations.

To compensate these shortcomings, a heuristic is required to re-index the new trajectory in time, and simultaneously optimize a measure of how well the new trajectory follows the objective one. This heuristic search is highly task-dependent and non-trivial and is less intuitive in high-dimensional spaces. One of these

Table 2.4: Approaches to Programming by Demonstration. PbD has been used for two main sets of applications, each one of which uses several metrics to evaluate how the techniques perform. Each approach is listed alongside its corresponding reference.

Applications	Metrics	Approaches	Related Works
Robot 3D position and velocity coordinates of end-effector and/or joint angles	RMS, RMSE, Learning Time, Norm of Jerk, Retrieval Duration, Max/Rel. Likelyhood, Stability, External Reward	HMM+GMR	Calinon et al. (2010a)
		DMP	Ijspeert et al. (2001)
		MoMP	Mülling et al. (2013)
		BM	Khansari-Zadeh and Billard (2010)
		SEDS	Khansari-Zadeh and Billard (2011)
		RL	Guenter and Billard (2007)
Robot position, orientation and velocity. Obstacles distances and discrete actions	Task Performance, Instructor Evaluation, Max. Reward Function	IRL	Abbeel and Ng (2004)
		RL	Nicolescu et al. (2008)
			Argall et al. (2011a)

time-dependent approaches (Coates et al., 2008) uses Expectation Maximization (EM) and an extended Kalman's filter to follow a given trajectory. This algorithm also learns a local model of the robot's dynamics along the chosen trajectory.

EM was also used in Dempster et al. (1977) to optimize a Gaussian Mixture Model (GMM) for the estimation of the parameters of existing models. In order to find a statistical noise-free estimation of the dynamic model several approaches have been proposed using either Gaussian Process Regression (GPR) (Williams and Rasmussen, 2006), Locally Weighted Projection Regression (LWPR) (Vijayakumar and Schaal, 2000) or Gaussian Mixture Regression (GMR) (Calinon et al., 2010a) were proposed. GMM and GPR find a locally optimal model of the function by maximizing the likelihood for a complete model to fit the data, while LWPR minimizes the RMSE between the estimates and the data. One of the main drawbacks of these approaches is that they cannot guarantee a stable estimate of the motion since there is no stability constraint forced near the optimization attractor point.

Dynamic Movement Primitives (DMP) (Pastor et al., 2009), originally proposed by Ijspeert et al. (2002b), is a method by which a non-linear dynamic model can be estimated and global stability at the optimization attractor point ensured: that is to

say, complex dynamics are robust and encoded precisely. DMP is also robust against perturbations and makes it possible to change parameters in the trajectory without altering the overall shape of the movement. These models are straightforwardly learned by imitation and well suited for reward-driven self-improvement. Mixture of Motor Primitives (MoMP) is an extension of these models recently proposed by Mülling et al. (2013) to cope with complex motor tasks requiring several movement primitives. MoMP creates a framework based on the idea that complex motor tasks can frequently be solved using a relatively small number of movement primitives and do not require a complex monolithic approach to cope with an entire task.

A different robust approach complementary to DMP is that of Stable Estimator of Dynamical Systems (SEDS) by Khansari-Zadeh and Billard (2011), which ensures time-independent learning generalized dynamics from multiple demonstrations. SEDS also outperforms Binary Merging (BM), previously proposed by Khansari-Zadeh and Billard (2010), in that it ensures globally asymptotic stability instead of local stability and better generalizes the motion for trajectories far from those in the demonstrations. BM is more accurate, more flexible and ensures that motion is locally stable. SEDS is more constrained because it fits a motion with a single globally stable dynamics. SEDS and DMP are complementary in the following way. DMP must be used whenever a motion is intrinsically time-dependent and only a single demonstration is available. In contrast, when the motion is time-independent and multiple demonstrations are available, SEDS would be the choice. A third time-independent approach based on Hidden Markov Models (HMM) and GMR has been proposed in Calinon et al. (2010a). This method evaluates the eigenvalues of each linear dynamic system and ensures that they all have negative real parts (stable). Nevertheless, asymptotic stability is not guaranteed.

For objectives such as position, orientation, and velocity of the robot, distances to the obstacles and a discrete set of actions, most of the techniques are reward-based: that is, a known reward function is assumed to guide the exploration ((Nicolescu et al., 2008; Grollman and Jenkins, 2010; Chernova and Veloso, 2009; Argall et al., 2011a,b)). The robot's policy will then consist of choosing actions that maximize

an expected reward function. In order to avoid this choice, Inverse Reinforcement Learning (IRL) (Abbeel and Ng, 2004) provides a framework for automatically determining the reward and discovers the optimal control policy using a discrete state action space. Alternative approaches derive a cost function in a continuous space (Ratliff et al., 2006, 2009). Recent IRL studies discuss multiple experts and identify multiple reward functions (Choi and Kim, 2012). The goal here is that this multiplicity of policies will make the controller more robust by offering alternative ways to complete the task whenever the context no longer allows the robot to perform the task in the optimal way.

More recently, (Grollman and Billard, 2011) proposed an approach based on learning from a set of failures in which the success of the learned task is represented by a binary value. This paper offers an interesting alternative to approaches that combine Imitation Learning and Reinforcement Learning since no reward function needs to be explicitly determined.

2.2 Action Recognition

Action recognition has become a very important topic in computer vision, with many fundamental applications, in robotics, video surveillance, human computer interaction, multimedia retrieval, biometrics based on gait or face, and hand and face gesture recognition among others. Today, the application to surveillance is natural in environments where the tracking and monitoring people is becoming an integral part of everyday activities. However, since our motivation relies on the aim of answering the question of *what to imitate?*, we concentrate on approaches that classifies full-body motions, such as boxing, kicking, bending, etc. and we categorize them according to how they represent the spatial and temporal structure of actions.

A significant amount of progress on human activity recognition has been made in the past 10 years, but it is still far from being an off the shelf technology. We are at a stage where experimental systems are deployed at airports and other public places. It is likely that more and more, such systems will be deployed.

Additionally, many approaches assume that the video is readily segmented into sequences that contain one instance of a known set of action labels. Often, it is also assumed that the location and approximate scale of the person in the video is known or can easily be estimated. The action detection task is thus ignored, which limits the applicability to situations where segmentation in space and time is possible. While several works (Hu et al., 2009; Yuan et al., 2009) have addressed this topic, it remains a challenge to perform action detection for online applications.

In this section we summarize the methodologies that have previously been explored for the recognition of human activities, and discus advantages and disadvantages of these approaches. An approach-based taxonomy is exposed and applied to categorize previous works.

2.2.1 Image Representations

In human action recognition, the common approach is to extract image features from the video and to issue a corresponding action class label. The classification algorithm is usually learned from training data.

As stated in Poppe (2010), image representations can be divided into two categories: local representations and global representations. On one hand, local representations describe the observation as a collection of independent patches. The calculation of local representations proceeds in a bottom-up fashion: spatio-temporal interest points are detected first, and local patches are calculated around these points. Finally, the patches are combined into a final representation. After initial success of bag-of-feature approaches, there is currently more focus on correlations between patches. Local representations are less sensitive to noise and partial occlusion, and do not strictly require background subtraction or tracking. However, as they depend on the extraction of a sufficient amount of relevant interest points, pre-processing is sometimes needed, for example to compensate for camera movements.

On the other hand, global representations encode the visual observation as a whole and they are obtained in a top-down fashion: a person is localized first in the image using background subtraction or tracking. Then, the region of interest is

encoded as a whole, which results in the image descriptor. The representations are powerful since they encode much of the information. However, they rely on accurate localization, background subtraction or tracking. Also, they are more sensitive to viewpoint, noise and occlusions. When the domain allows for good control of these factors, global representations usually perform well.

Considering the occlusions issue, although global image representations have proven to yield good results, and they can usually be extracted with low cost, their applicability is limited to scenarios where Region of Interests (ROI) can be determined reliably. Moreover, they cannot deal with occlusions. These issues are addressed with local representations. Initial work used bag-of-feature representations but more recent work takes into account spatial and temporal correlations between patches. Still, the question how to deal with more severe occlusions has largely been ignored.

With respect to the point of view, most of the reported work is restricted to fixed and known viewpoints, which severely limits its applicability. The use of multiple view-dependent action models solves this issue but at the cost of increased training complexity. Recently, researchers have begun to address the recognition of actions from viewpoints for which there is no corresponding training data (Farhadi and Tabrizi, 2008; Farhadi et al., 2009; Junejo et al., 2008).

Further, today's environment for human activity recognition is significantly different from the scenario at the end of the last decade. The cameras were mostly fixed cameras and without pan-tilt-zoom adjustments. Today's cameras may be mounted on several types of moving platforms ranging from a moving car or a truck to an Unmanned Aerial Vehicles (UAV). A global positioning system may be attached to the camera system to pin-point its location. The recognition of activity from a moving platform poses many more challenges. Noise, tracking, and segmentation issues arising out of stabilization of video add to the difficulty of the problem of the recognition of activities. Tracking is a difficult problem though animals and human do it almost effortlessly. If the tracking algorithm does not extract the object of the focus of attention, recognition of the activity being performed becomes

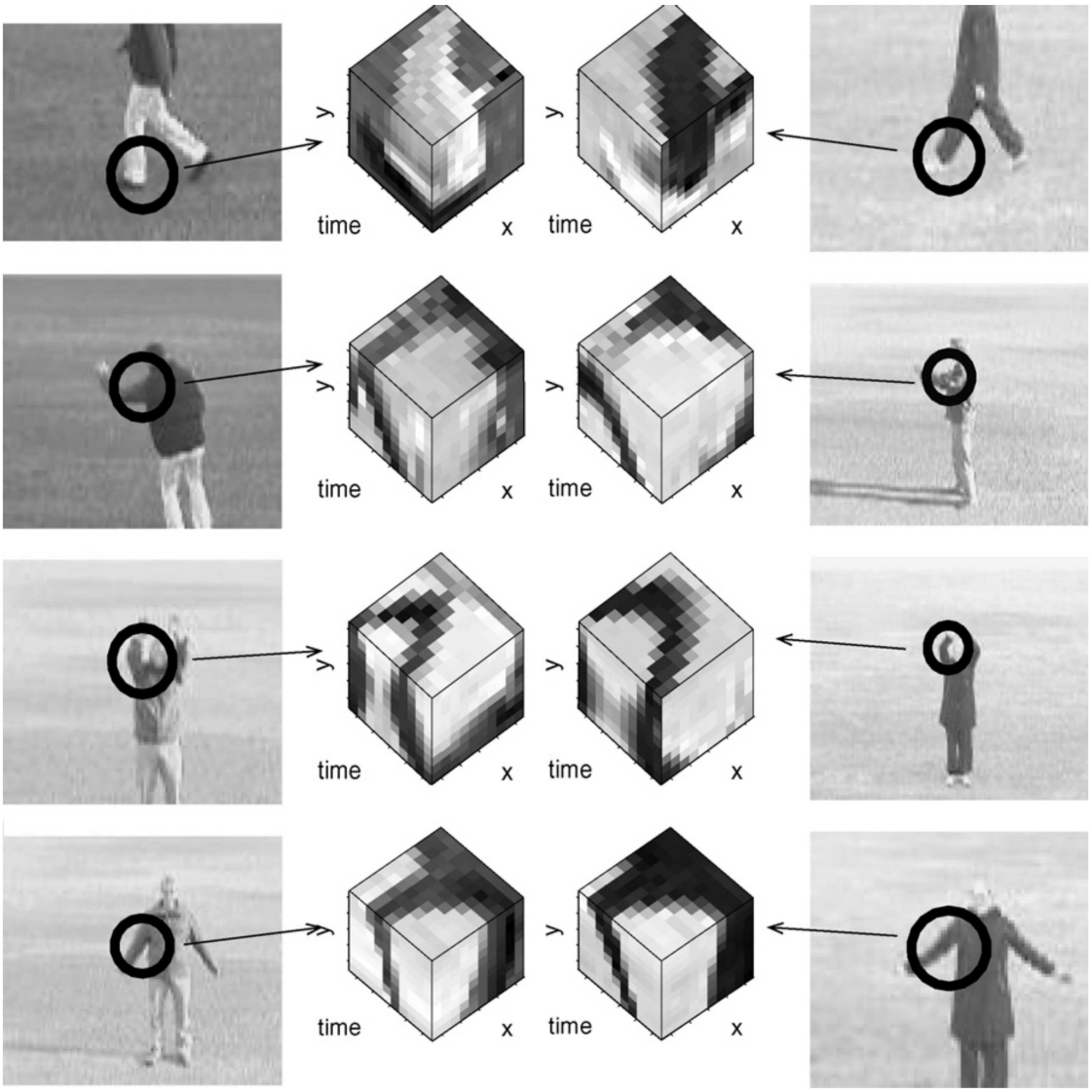


Figure 2.1: Extraction of spacetime cuboids at interest points from similar actions performed by different persons (reprinted from Laptev et al. (2007), ©Elsevier, 2007).

enormously more difficult. Designing an activity recognition system which is able to compensate for such low-level failures in those environments (i.e . moving platforms) is an extremely challenging task.

Alternative approaches to segmenting body parts based on analyzing 3D XYT volumes by extracting gross features are being developed. In particular, 3D local patch features described in terms of Histogram of Gradients (HOG) and/or Histogram of Optical Flow (HOF), such as cuboids (Dollar et al., 2005) and 3D SIFT (Scovanner et al., 2007), are gaining popularity. A representation of these cuboids is shown in Figure. 2.1. These approaches are motivated by the success of

object recognition using 2D local descriptors (e.g. Scale Invariant Feature Transform (SIFT) (Lowe, 1999)).

However, they involve long feature vectors obtained from a large 3D XYT volume created by concatenating image frames, and are likely to have an impact on real time analysis. The 3D search space is much larger than its 2D versions. Further, the existing local space-time features generally require a non-textured background for reliable recognition, such as the ones in the KTH (Schuldt et al., 2004) and Weizmann (Blank et al., 2005) databases. Also, a limited amount of work has been published on the 3D feature-based approaches for analysis of complex human activities. What one needs is an approach which exploits the easy computation of SIFT, HOG, and HOF operators and avoids the difficulties of segmentation of body parts and/or combines the two approaches in a meaningful way.

2.2.2 Classification

There exists direct classification and temporal state-space models. In the former, temporal variations are not explicitly modeled, which proved to be a reasonable approach in many cases. For more complex motions, it is questionable whether this approach is suitable. Generative state-space models such as Hidden Markov Models (HMM) can model temporal variations but have difficulties distinguishing between related actions (e.g. jogging and walking). In this respect, discriminating graphical approaches are more suitable. Thus, the flexibility of the classifier with respect to adding or removing action classes from the repertoire will play a more important role.

2.2.3 Real-time Applications

One promising direction for enabling real-time implementation is the study of hardware supports. Rofouei et al. (2008) have implemented a GPU-based version of the cuboid feature extractor, utilizing Graphical Processing Unit (GPU) with tens of cores running thousands of threads. The GPU-version turned out to be 50 times faster than the CPU counterpart of it, while obtaining the same results. Modern

CPUs and GPUs are composed of multiple cores, and the number of cores is likely to continually increase for the next few years, suggesting computer vision researchers to explore the utilization of them.

2.2.4 Databases

The use of publicly available databases allows for the comparison of different approaches and gives insight into the abilities of respective methods.

2.2.4.1 Collection of Videos

Publicly available databases have shaped the domain by allowing for objective comparison between approaches on common training and test data. They also allow for better understanding of methods since researchers are aware of the challenges of each set. However, algorithms may be biased to a particular database. This may lead to complex approaches that perform better on a specific database but may be less generally applicable.

Also, given the increasing level of sophistication of action recognition algorithms, larger and more complex databases should direct research efforts to realistic settings. Initially, databases were not focused on an application domain. However, action recognition in surveillance, human-computer interaction and video retrieval poses different challenges. Human-computer interaction applications require real-time processing, missed detection in surveillance are unacceptable and video retrieval applications often cannot benefit from a controlled setting and require a query interface (e.g. (Suma et al., 2008)). Currently, there is a shift towards a diversification in databases. The HOHA database (Laptev et al., 2008) targets action recognition in movies, whereas the UFC sport database (Rodriguez et al., 2008) contains sport footage. Such a diversification is beneficial as it allows for realistic recording settings while focusing on relevant action classes. Moreover, the use of application-specific database allows for the use of evaluation metrics that go beyond precision and recall, such as speed of processing or detection accuracy. Still, the compilation or recording of database that contain sufficient variation in

Table 2.5: The human activity recognition categorization for non-hierarchical approaches.

Human activity recognition. Non-hierarchical. Single-layered.			**Related Works**
Space-time	Space-time volume	Template matching	Bobick and Davis (2001); Shechtman and Irani (2005); Rodriguez et al. (2008)
		Neighbor-based (discriminative)	Ke et al. (2007)
		Statistical modeling	-
	Trajectories	Template matching	Campbell and Bobick (1995); Rao and Shah (2001)
		Neighbor-based (discriminative)	Yilmaz and Shah (2005)
		Statistical modeling	Sheikh et al. (2005); Khan and Shah (2005)
	Space-time features	Template matching	Zelnik-Manor and Irani (2001); Laptev (2005)
		Neighbor-based (discriminative)	Schuldt et al. (2004); Dollar et al. (2005); Blank et al. (2005); Laptev et al. (2008); Ryoo and Aggarwal (2009b)
		Statistical modeling	Chomat and Crowley (1999); Niebles et al. (2008); Wong et al. (2007); Lv et al. (2004)
Sequential	Exemplar-based		Darrell and Pentland (1993); Gavrila et al. (1995); Yacoob and Black (1998); Efros et al. (2003); Lublinerman et al. (2006); Veeraraghavan et al. (2006); Jiang et al. (2006); Vaswani et al. (2003)
	State-based		Yamato et al. (1992); Starner and Pentland (1997); Bobick and Wilson (1997); Oliver et al. (2000); Park and Aggarwal (2004); Natarajan and Nevatia (2007); Moore et al. (1999); Peursum et al. (2005); Gupta and Davis (2007)

movements, recording settings and environmental settings remains challenging and should continue to be a topic of discussion.

2.2.4.2 Labels

Related is the issue of labeling data. For increasingly large and complex databases, manual labeling will become prohibitive. Automatic labeling using video subtitles (Gupta and Mooney, 2009) and movie scripts (Cour et al., 2008; Duchenne et al., 2009; Laptev et al., 2008) is possible in some domains, but still requires manual verification. When using an incremental approach to image harvesting such as in Ikizler-Cinbis et al. (2009), the initial set will largely affect the final variety of action performances. We discussed vision-based human action recognition in this survey but a multi-modal approach could improve recognition in some domains, for example

Table 2.6: The human activity recognition categorization for hierarchical approaches.

Human activity recognition. Hierarchical.		**Related Works**
Statistical	Human actions	Nguyen et al. (2005)
	Human- human interactions	Oliver et al. (2002)
	Human- object interactions	Shi et al. (2004); Yu and Aggarwal (2006); Damen and Hogg (2009)
	Group activities	Cupillard et al. (2002); Gong and Xiang (2003); Zhang et al. (2004); Dai et al. (2008)
Syntactic	Human actions	-
	Human- human interactions	Ivanov and Bobick (2000); Joo and Chellappa (2006)
	Human- object interactions	Moore and Essa (2002); Minnen et al. (2003)
	Group activities	-
Description-based	Human actions	Pinhanez and Bobick (1998); Gupta et al. (2009)
	Human- human interactions	Intille and Bobick (1999); Vu et al. (2003); Ghanem et al. (2004); Ryoo and Aggarwal (2009a)
	Human-object interactions	Siskind (2001); Nevatia et al. (2003, 2004); Ryoo and Aggarwal (2007)
	Group activities	Ryoo and Aggarwal (2008)

in movie analysis. Also, context such as background, camera motion, interaction between persons and person identity provides informative cues (Marszalek et al., 2009).

2.2.5 Taxonomy of Action recognition

Following the work of Aggarwal and Ryoo (2011) the Action Recognition approaches can be categorized into non-hierarchical approaches developed for the recognition of gestures and actions as well as hierarchical approaches for the analysis of high-level interactions between multiple humans and objects. Non-hierarchical approaches are again divided into space-time approaches and sequential approaches, and the similarities and differences of the two approaches are discussed thoroughly. Additionally, previous publications following statistical, syntactic, and description-based approaches for hierarchical approaches are compared.

Hierarchical recognition approaches are being studied intensively especially for the recognition of complex multi-person activities. Particularly, description-based approaches are gaining an increasing amount of popularity because of their

ability to represent and recognize human interactions with complex spatio-temporal structures. Activities with structured scenarios (e.g. most of surveillance scenarios) require hierarchical approaches, and they are showing the potential to make a reliable decision probabilistically. Hierarchical approaches have their advantages in recognition of high-level activities performed by multiple persons, and they must be explored further in the future to support demands from surveillance systems and other applications. Both Tables 2.5 and 2.6 summarizes this categorization and include some of the most representative studies of the state of the art for each category.

Given the current state of the art and motivated by the broad range of applications that can benefit from robust human action recognition, it is expected that many of these challenges will be addressed in the near future. This would be a big step towards the fulfillment of the longstanding promise to achieve robust automatic recognition and interpretation of human action.

Chapter 3

Analyzing Bag of Words

"Pain is temporary, glory is forever"

- Anonymous,

3.1 Outline of the Chapter

In computer vision, action recognition is a common topic of the State of the Art. Bag of Visual Words (BoVW) method has been recently widely used for this topic. The principal point of this chapter is to show the influence of parameter variation in the traditional BoW approach for the three phases in which can be divided: first, the interest points detection and descriptor extraction, second, the codebook generation, and third, the pooling and classification phase. Specifically, we pay special attention in varying methods for clustering information extracted from the image, i.e. to build a good codebook, because the number of clusters has high influence over the results and it should be estimated by the system. The chapter is organized as follows:

- Section 3.2 introduces the problem and the related work existing on this topic.
- Section 3.3 show the influence of proper interest points detection and descriptor extraction.
- Section 3.4 show the influence of proper codebook generation.
- Section 3.5 show the influence of proper pooling and classification.
- Section 3.7 summarizes the contribution of the approach.

3.2 Introduction

Action recognition is a very active research topic in computer vision with many important applications, including video surveillance, human computer interaction, robotics and programming by demonstration among others. Action recognition is the process of naming actions, usually as an action verb. To reach that goal, many approaches typically make use of a combination of vision and machine learning techniques. Vision techniques try to extract action features from the videos, while machine learning techniques try to learn statistical models from those features, and classify new features using the learned model. A wide range of Action Recognition methods exists and they can be classified by spatial or temporal representations, as we showed previously in Chapter 2.

In our approach we make use of a common approach called spatio-temporal Bag of Words representation. In this method, spatial and temporal information are extracted from the surroundings of an Interest Point (IP) in order to build the feature descriptor. From this set of features a dictionary is computed and quantized to represent each snippet of the video. To learn a model and further classification of new features, a Support Vector Machine (SVM) is employed.

Related Work

Authors who make use of BoW approach, use to vary the three main phases: feature extraction, vector quantization and pooling. In Laptev et al. (2008) an extended Harris corners was employed to detect Spatio Temporal Interest Points (STIP), to compute a descriptor composed by a Histogram of Gradients (HOG) concatenated with a Histogram of Optical Flow (HOF) (HOG–HOF descriptor). They used a K-means algorithm to cluster the features and a k-Nearest Neighbor (kNN) with euclidean distance for pooling. Finally, a SVM with χ^2 kernel was used to learn a model and classify new instances. A novel approach were proposed in Wang et al. (2011), in which they built a dense grid of Interest Points and extracted a

trajectory based descriptor (Dense trajectories). More recently, Zhang et al. (2012) made use of sparse code to create the dictionary, a variant of the K-means clustering algorithm where they tuned the constraint with a λ parameter in order to make it less restrictive.

3.3 Interest Points Detection and Descriptor Extraction

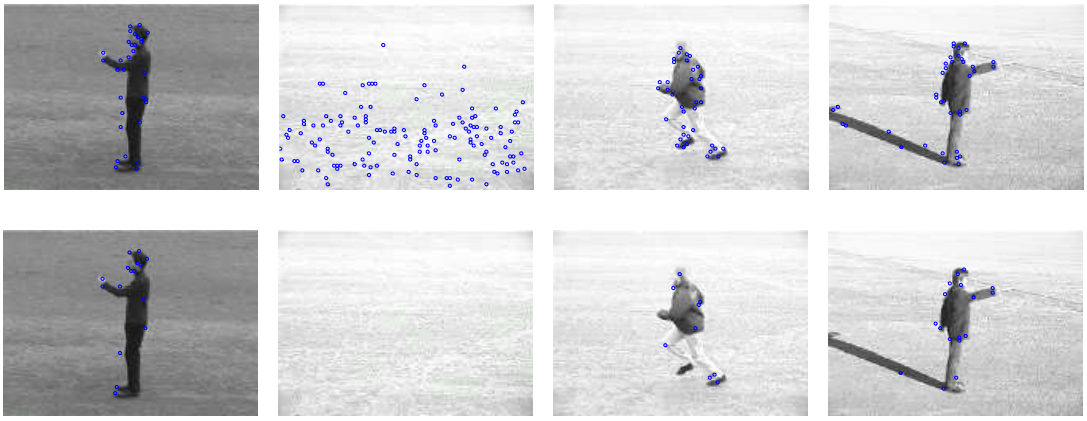


Figure 3.1: Interest Points from Harris corner detector for database frames. First row are the IPs detected. Second row are the IPs randomly selected before clustering.

We extract Harris corners as Interest Points (IPs) in a similar way as it is done in Laptev et al. (2008). Sample action frames from the database are shown in Figure 3.1 in two rows: the top row shows frames with the detected corners, and the second row shows the selected points that are going to be clustered. For a sampling point s, position coordinates (x_s, y_s) are kept to which are added to them the third dimension, which is the time (t_s), i.e., the frame number, and the spatial and temporal scale (σ_s, τ_s), both used to determine at which scale the descriptor is computed. We use 3 Dimensional Histogram of Gradients (HOG3D) descriptor (Kläser et al., 2008) to code the characteristics of these IPs. Histogram of Gradients have been widely used for object recognition in static images, but, in our approach, we need a descriptor which can relate spatial and temporal information. HOG3D is similar to HOG descriptor, in the sense that it computes the gradient histograms of

a pixel surroundings, but with the main advantage that it generalizes the concept to 3D. The final descriptor d_s for s is computed for a local support region r_s with a width (w_s), height (h_s) and length (l_s) around the position s, given by $w_s = h_s = \sigma_0 \cdot \sigma_s$ and $l_s = \tau_0 \cdot \tau_s$, where σ_0 and τ_0 parameters characterize the relative size of the support region around s.

This local support region r_s is divided into a set of $MxMxN$ cells c_i. For each cell, an orientation histogram is computed by $h_c = \sum_{i=1}^{S^3} q_{b_i}$, where q_{b_i} is the quantization employing a regular polyhedron for the subblock b_i. Finally, all histograms are concatenated to one feature vector $d_s = (d_1, ..., d_{M^2N})^T$. Final dimension of the feature can be pre-calculated by $dim\{d_s\} = M^2 \cdot N \cdot n$, with n the number of orientations, taking into account its full or half orientation. The relevance of this value lies in the computation time when clustering, which considerably increases as higher this dimension is.

3.4 Codebook Generation

The vast majority of proposals in this topic use a supervised clustering for codebook generation. We analyze this procedure, which has been widely seen that it has high influence to the final recognition performance. We know that the recognition performance steadily grows with the size of the codebook, as observed, e.g. by Csurka et al. (2004). To this purpose we compare three different methods. Firstly, we use standard K-means (MacQueen, 1967), which is the most common clustering algorithm for this topic. In this algorithm a K-value is needed to be provided as the final number of words representing all data collected from videos. A principal disadvantage of standard K-means is that clusters can only be separated by a hyper-plane. Using a weighted kernel K-means (Dhillon et al., 2007), nonlinear separators can be obtained. Secondly, we propose to use Meanshift (Comaniciu and Meer, 2002), in which the final number of clusters is not previously determined. With Meanshift approach, the algorithm determines itself the codebook size by just tuning a bandwidth parameter. As the third option, we propose the random selection of

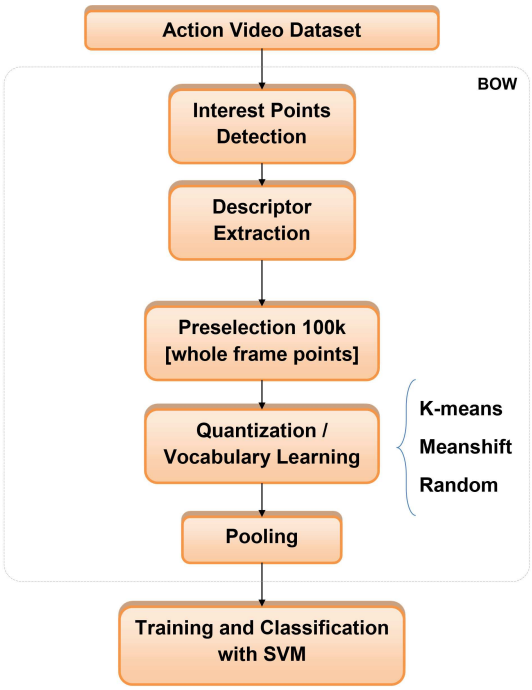


Figure 3.2: Flow chart of the methodology used to evaluate the three proposed clustering methods for action recognition.

the cluster centers. In the end, following BoW procedure, we find the word which represents each snippet and we use a non-linear SVM classifier to classify the videos.

The method we propose can be seen in Figure 3.2. We firstly extract Interest Points from video frames with Harris detector. Secondly, we compute the HOG3D descriptor for each IP. To build the codebook, and due to the computing complexity, we limit the amount of data to 100000 samples, and then it is computed by a clustering algorithm, which gives a dictionary with a specific number of words. We code, then, each snippet by a codebook word. Finally, a Support Vector Machine classifier is used to evaluate the performance of the recognition, which is used as the evaluation metric.

Traditionally, in action recognition, the codebook is generated by previously setting the number of words that it is desired to have. In this sense, K-means clustering algorithm gives quite good results. The intention of this work is to show why K-means seems to be better for that purpose than algorithms that can decide the number of clusters by themselves or selecting the clusters randomly.

3.4.1 K-means

Given a set of vectors, the K-means algorithm seeks to find clusters that minimize the objective function:

$$D(\{\pi_c\}_{c=1}^{k}) = \sum_{c=1}^{k} \sum_{a_i \in \pi_c} \|a_i - m_c\|^2 \tag{3.1}$$

$$\text{where } m_c = \frac{\sum_{a_i \in \pi_c} a_i}{|\pi_c|}$$

The centroid (or the mean) of the cluster π_c is denoted by m_c. A principal disadvantage of standard K-means is that clusters can only be separated by a hyper-plane.

3.4.2 Meanshift

The Meanshift algorithm is a non-parametric clustering technique which does not require prior knowledge of the number of clusters, and does not constraint the shape of the clusters. Given n data points x_i, $i = 1, .., n$ on a $d-$dimensional space R^d, the multivariate kernel density estimate obtained with kernel $K(x)$ and window radius h is:

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{3.2}$$

For radially symmetric kernels, it suffices to define the profile of the kernel $k(x)$ satisfying

$$K(x) = c_{k,d} k(\|x\|^2) \tag{3.3}$$

where $c_{k,d}$ is a normalization constant which assures $K(x)$ integrates to 1. The modes of the density function are located at the zeros of the gradient function $\nabla f(x) = 0$. The gradient of the density estimator is

$$\nabla f(x) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (x - x_i) g(\|\frac{x - x_i}{h}\|^2) \tag{3.4}$$

where $g(s) = -k'(s)$. The first term is proportional to the density estimate at x computed with kernel $G(x) = c_{g,d}g(\|x\|^2)$ and the second term

$$m_h(x) = \frac{\sum_{i=1}^{n}(x)g(\|\frac{x-x_i}{h}\|^2)}{\sum_{i=1}^{n} g(\|\frac{x-x_i}{h}\|^2)} \tag{3.5}$$

is the Meanshift. The Meanshift vector always points towards the direction of the maximum increase in the density. The Meanshift procedure, obtained by successive computation of the Meanshift vector $m_h(x^t)$, and translation of the window $x^{t+1} = x^t + m_h(x^t)$, is guaranteed to converge to a point where the gradient of density function is zero.

The Meanshift clustering algorithm is a practical application of the mode finding procedure: starting on the data points, run Meanshift procedure to find the stationary points of the density function, and prune these points by retaining only the local maxima. The set of all locations that converge to the same mode defines the basin of attraction of that mode. The points which are in the same basin of attraction is associated with the same cluster.

3.4.3 Random centers selection

When the amount of features extracted from videos is big enough, e.g. 100000 or more, it is reasonable to select the centers of each cluster randomly. However, due to its randomness, non-representative selection can occur. This issue is solved by iterating the selection after the classification stage by using Random Sample Consensus (RANSAC) (Fischler and Bolles, 1981).

3.5 Pooling and Classification

We base our approach to the traditional Bag of Visual Words (BoVW), building a codebook and representing videos with words of this dictionary. In this work we limit the computation complexity using a subset of 100000 uniformly selected samples to construct the codebook.

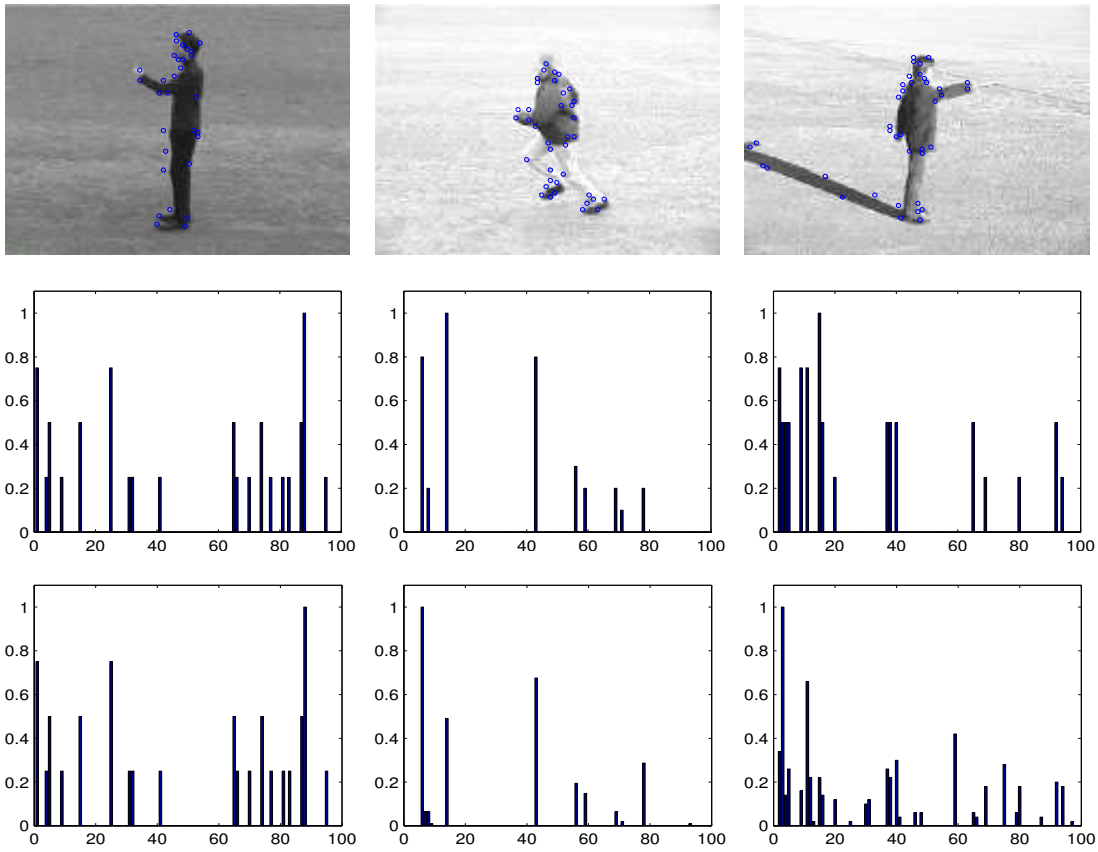


Figure 3.3: Codebook words present in each action. First row: Interest points for frame extracted with Harris corner detector. Second row: histogram of codebook words appearances in each frame. This histogram, when normalized, is the image descriptor. Third row: cumulative histogram of frequencies from first frame to the one shown in the first row.

We use a non-linear SVM classifier to recognize actions with a χ^2 kernel likewise Laptev et al. (2008),

$$K(v_i, v_j) = exp(D(v_i, v_j)) \tag{3.6}$$

where $D(v_i, v_j)$ is the χ^2 distance between video v_i and v_j. Since our action recognition is the multi-class classification, we use a one-against-all approach and determine the class with the highest confidence score. The Figure 3.3 represents the frame coding through BoW approach. The histograms from the second row, when normalized, are the frame descriptors.

3.6 Experimental Results and Discussion

We performed the experiments with a 8x i7-2600 CPU at 3.40GHz and developed a visualization tool in which we could analyze the influence of the codebook words over each frame of each video and over the entire sequence. The BoW descriptor of each image is visualized with this tool as can be seen in Figure 3.4. Following we summarizes of what is exposed in this results section: in Section 3.6.1 we present a brief description of the dataset used for validating our method. In Section 3.6.2 the results obtained for each proposed clustering method are compared, using the final recognition performance as a metric. Then, the influence of the codebook size is evaluated in Section 3.6.3. Finally, the selection of the best kernel used for the classifier is studied in Section 3.6.4.

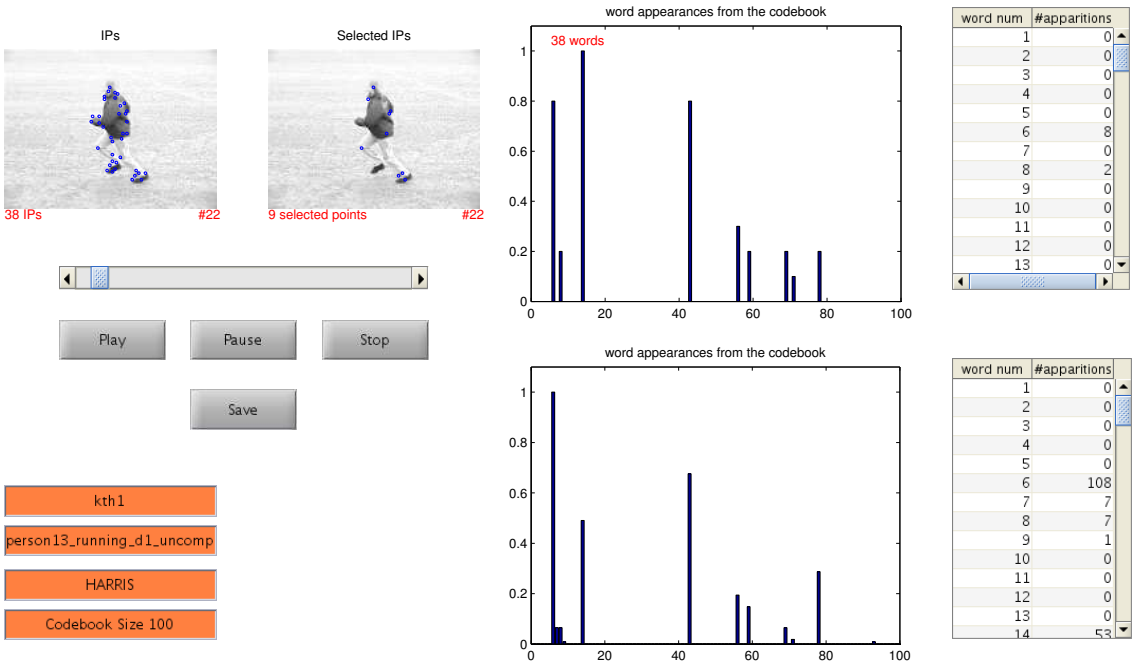


Figure 3.4: Visualization tool. With this tool, every frame BoW coding and the cumulative BoW for the video are visible as well as the values of each histogram bin in the most right tables. As can be seen in the left lower part of the visualization tool, dataset name, video name, IP detector used and size of the codebook are shown.

Figure 3.5: KTH Dataset: boxing, hand waving and running are used in our experiments.

3.6.1 Datasets

The KTH dataset (Schuldt et al., 2004) contains six different human actions: *boxing*, *hand waving*, *hand clapping*, *walking*, *jogging* and *running*. Each action is performed by 25 subjects in 4 different scenarios. In our experimental setup, we only use three of the six actions (boxing, hand waving and running) performed by randomly selected performers. We evaluate a SVM classifier for each of these actions using a one-against-all cross-validation. In Figure 3.5 we show the three actions taken from the dataset.

3.6.2 Comparison between Methods

In our experiments, we follow the setup of Kläser et al. (2008), in which the HOG3D parameters are optimized for KTH Dataset. We extract Harris corners as IP due to these kind of points have shown better performance compared to DENSE sampling, Features from Accelerated Segment Test (FAST) or Speeded Up Robust Features (SURF) interest points, as can be seen in Table 3.1. Using a K-means codebook of 4000 words, DENSE sampling with a 9x9 grid gives 82.04% mean performance of action recognition, better than FAST points extraction, with 81.66%. However, the best results are obtained with Harris IPs, with 91.50% of accuracy.

Table 3.1: Comparison between IPs extraction methods using HOG3D descriptor. The values are for a codebook size of 4000 words and linear kernel of the SVM.

Method	No action	Boxing	Handwaving	Running	Mean
DENSE 9x9	83.3%	83.0%	79.4%	83.7%	82.04%
Harris	93.0%	94.1%	92.6%	87.8%	91.50%
FAST	87.3%	81.9%	82.1%	80.9%	81.66%
SURF	83.3%	78.8%	79.1%	77.9%	79.77%

For codebook generation, we set the number of words to 1000 and 4000, for

K-means and random selection, and we have tuned the Meanshift bandwidth accordingly to the number of clusters desired, i.e., around 4000 words. This bandwidth value is set to 3. For K-means we guarantee the minimum clustering error by setting the iterations to 10, and for random selection we iterate also 10 times to guarantee the best random sampling.

As it is shown in Table 3.2, Meanshift clustering takes the longest time to finally give similar results to others. It can be seen also that despite the variation in the number of words, K-means always outperforms random selection.

Table 3.2: Comparison between codebook generation methods: K-means, Meanshift and random selection. Values obtained by using a Harris corner detector and HOG3D descriptor, with a SVM classifier.

Cluster	n° clusters	n° iterations	performance	computation time (s)
K-means	1000	10	83.4%	689
K-means	4000	10	91.5%	4656
Meanshift	3319	-	89.8%	42156
random	1000	10	79.4%	0
random	4000	10	90.3%	0

3.6.3 Codebook Size Influence

In order to see the influence of the codebook size, we compute nine different dictionaries sizes for K-means and random, of 100, 500, 1000, 1500, 2000, 2500, 3000, 3500 and 4000 words. Meanshift is used once, with a bandwidth of 3. This is due to we want a significant number of clusters, i.e. 4000. With this bandwidth value we get 3319 clusters.

As can be seen in Figure 3.6, we need a large amount of words to get a good recognition. This means that clustering process is so crucial, and it usually takes a substantial quantity of time to compute this dictionary.

3.6.4 Kernel Selection

As soon as we decide to use the kernel trick for the SVM classifier, we discover that there exists numerous possibilities that are good candidates to be a kernel

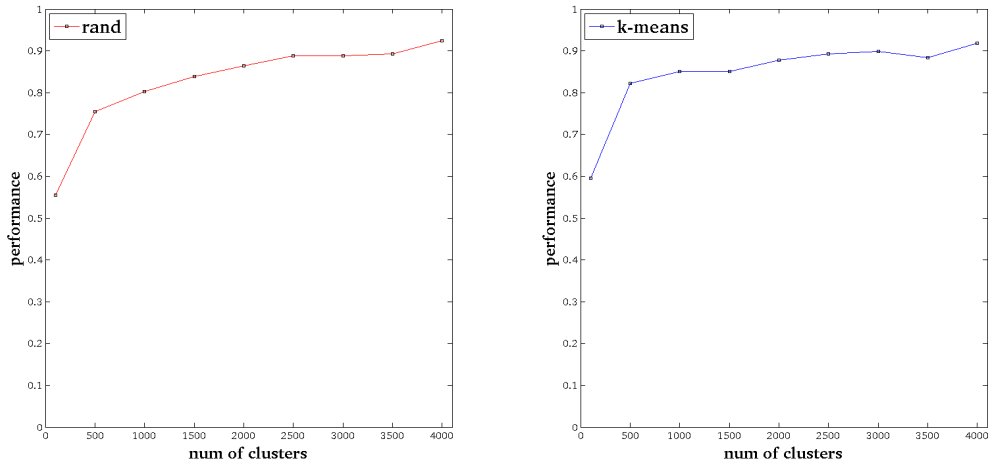
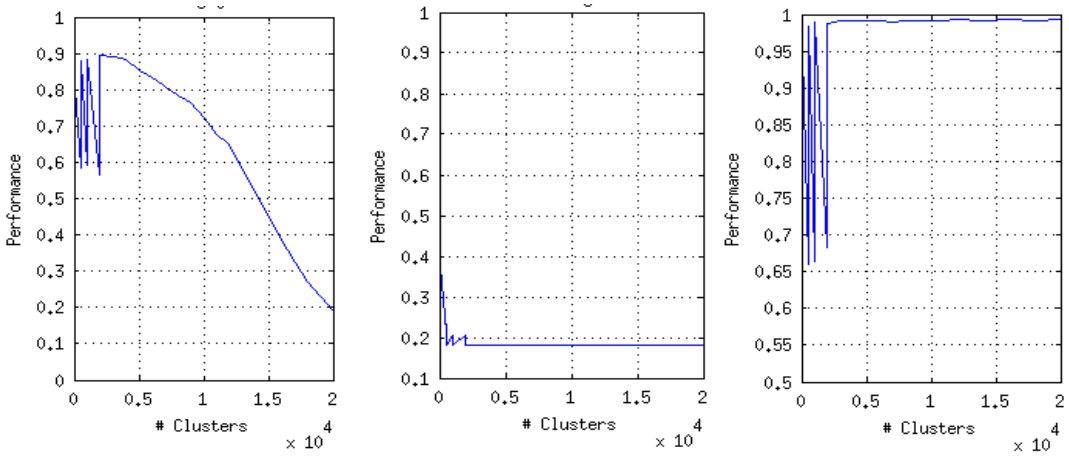


Figure 3.6: Codebook size influence. Results obtained by using Harris corner detection, HOG3D descriptor and SVM classifier. The higher the number of words belonging to the dictionary the better the performance will be.

function. We only have to ensure that the function applied to two vectors u and v is positive definite. Hence, linear kernel $k(x, y) = u' * v$ is the most common function used by the scientific community. However, non linear functions can be used, like polynomials $k(x, y) = (gamma * u' * v + coef0)^degree$, RBF $k(x, y) = exp(-gamma * |u - v|^2)$, sigmoid $k(x, y) = tanh(gamma * u' * v + coef0)$, chi-squared $k(x, y) = 1 - 2 * sum((xi - yi)^2/(xi + yi))$ or exponential chi-squared $k(x, y) = exp(-gamma * sum((xi - yi)^2/(xi + yi)))$. Results for all these kernels are presented in Figure 3.7. The results show that linear functions does not have constancy with the increase of the codebook size, with a decrease of the performance starting from 1500 words. However, this curse of dimensionality can be avoided by using chi-squared kernels. With these functions, the performance is even higher than any other kernel functions type, reaching results close to the 100% with small codebook sizes. The polynomial, RBF and sigmoid functions have shown not to be adequate to solve action recognition problems with the traditional BoW.

3.7 Summary

In this chapter we have analyzed and compared three different clustering methods to compute the codebook of the traditional BoW approach. We have used the final



(a) linear (b) RBF (c) chi-squared

Figure 3.7: Kernel selection for the SVM classifier. The figures are obtained by using a dense grid selection of IP with a HOG3D descriptor, a k-means clustering method and a SVM classifier.

action recognition performance as the evaluation metric to carry out our purpose and, finally, we have evaluated our framework on a public action database. We have discussed the importance of the clustering parameters selection to determine their influence over the recognition results. In the end, we proposed to take into account the random selection, by which a surprisingly good recognition performance is achieved. In the next chapter, we will build a semantic relationship between objects and actions, and we will need an efficient clustering method that be fast and good enough for real-time applications.

Chapter 4

Context Information

"When you see a good move, look for a better one"

- Emanuel Lasker,

4.1 Outline of the Chapter

Classifying web videos using a Bag of Words (BoW) representation has received increased attention due to its computational simplicity and good performance. The increasing number of categories, including actions with high confusion, and the addition of significant contextual information has lead to most of the authors focusing their efforts on the combination of descriptors. It is widely accepted that using descriptors that give different kind of information tends to increase the performance. In this field, we propose to use the multikernel Support Vector Machine (SVM) with a contrasted selection of kernels introducing contextual information, i.e. objects directly related to performed action by pre-selecting a set of points belonging to objects to calculate the codebook. In order to know if a point is part of an object, these items are previously tracked by matching consecutive frames, and the bounding box is calculated and labeled. We code the action videos using BoW representation with the object codewords and introduce them to the SVM as an additional kernel. Experiments have been carried out on two action databases, KTH and HMDB, the results provide a significant improvement with respect to other similar approaches.

The chapter is organized as follows:

- Section 4.2 introduces the problem and the related work existing on this topic.
- Section 4.3 presents the inclusion of context to the traditional BoW approach making use of objects present in the action performed.
- Section 4.4 presents the experimentation and discussion.
- Section 4.5 summarizes the contribution of the approach.

4.2 Introduction

The number of videos uploaded online is increasing every day and recently the analysis of their content has become an intense field of research. In this context, our research focuses on the recognition of action in videos containing contextual information about the means by which an action is carried out. Initially, the sort of databases over which the actions were recognized conformed a set of videos where scenes and parameters such as illumination, focus, distance, and viewpoints were mostly controlled, and few or none data existed on the tools and objects that were involved in the action. For example, the KTH database (Schuldt et al., 2004), a popular choice to test different action recognition techniques, has not such kind of information. In any case, we use this database in the present work to show the performance of our approach in comparison to the rest of other state-of-the-art approaches.

Recently, however, more realistic databases have increasingly been employed in order to go beyond the current state of the art. These sets include videos that stage more realistic actions. A relevant database, HMDB (Kuehne et al., 2011), is one of the largest action video database to-date with 51 action categories, which in total contain almost 7,000 manually annotated clips extracted from a variety of sources ranging from digitized movies to YouTube. This database has been created to evaluate the performance of computer vision systems for action recognition and explore the robustness of these methods under various conditions such as cluttered backgrounds, fast irregular motions, occlusions and camera motion.

Several approaches have been proposed in the literature for the recognition of actions in diverse real-world videos. In this database, actions that are contextually connected to the tool or object employed in their performance can be found.

In order to increase the robustness of the recognition of actions in more challenging situations, we propose an approach that is able to integrate two sources of information relevant to discriminate actions, namely, the space-time data that describes the motion and the contextual information that explains how the action is carried out. Specifically, by using the HMDB (Kuehne et al., 2011) database, we select a subset of actions that are performed using a tool or object, a contextual information that allows to discriminate apparently similar actions such as shooting a gun or a bow, which its biggest difference lies in the object employed to carry out the action. We explain how these different sources of information can be combined in richer description of human actions that permits higher recognition rates.

The main contributions of this chapter is dual: first, the introduction of contextual information of actions into BoW-based descriptors, and second, a recognition structure that allows the addition of new information using multichannel SVM (Zhang et al., 2006). Multichannel SVM has previously proven very successful in action recognition (Wang et al., 2013; Bilinski and Corvee, 2013) and we take advantage of this structure by adding data which is not strictly a descriptor of motion but contextual information describing the tool employed in the action, which is a new way of using multichannel SVM.

Related Work

What concerns the type of action descriptor, local space-time features (Dollar et al., 2005; Laptev, 2005) have shown to be successful for general action recognition because they avoid non-trivial pre-processing steps, such as tracking and segmentation, and provide descriptors invariant to illumination and camera motion. In particular, HOG3D (Kläser et al., 2008) has proven to outperform most of this sort of descriptors (Willems et al., 2008; Scovanner et al., 2007). Another approach

has been trying to find the best combinations of different simpler descriptors. To this end, Snoek et al. (2005) studied the different methods of descriptor fusion and classified them into early or late fusion approaches. The former one consists in a fusion before the training step, while the latter is a fusion afterwards.

With respect to the combination of features, Ikizler-Cinbis and Sclaroff (2010) combined six different descriptors for three different contextual information, namely, *people* (HOG and HOG3D), *objects* (HOF and HOG), and *scene* (GIST and color histograms). Their combination is accomplished by using a multiple Multiple Instance Learning (MIL) approach, which is a concatenation of bag representations and classified with an L2-Regularized Linear SVM. On the other hand, Bilinski and Corvee (2013) used relative dense tracklets for action recognition. They computed two specific descriptors, Shape Multi Scale Tracklet (SMST) and Relative Multi Scale Tracklet (RMST), in order to obtain information from the actions relative to the head of the performer. Two more descriptors encoding space and time, HOG and HOF, were employed. A multichannel χ^2 kernel SVM was used for the combination of this set of descriptors. Similarly, Wang et al. (2011) and later Wang and Schmid (2013) computed dense trajectories and their descriptors –Histogram of Gradients (HOG), Histogram of Optical Flow (HOF), and Motion Boundary Histogram (MBH)– to finally combine them using a multichannel SVM. In contrast, using a late fusion of the descriptors the approach of Reddy and Shah (2013) trained a SVM for a scene context descriptor and another SVM for a motion descriptor, using a histogram intersection kernel. The two probability estimates obtained separately from each SVM were fused into one single recognition output afterwards.

In the work presented in this chapter, we use information describing the object involved in an action using a BoW based action recognition approach. To this end, we first detect the set of points belonging to the object by matching them to an instance of the object. This process also labels the bounding boxes, which are later used to compute a new codebook –the dictionary employed to compute the relative frequencies in a BoW description–, and the information about the objects in the actions is preserved as a consequence. Afterwards, we employ such codebook to

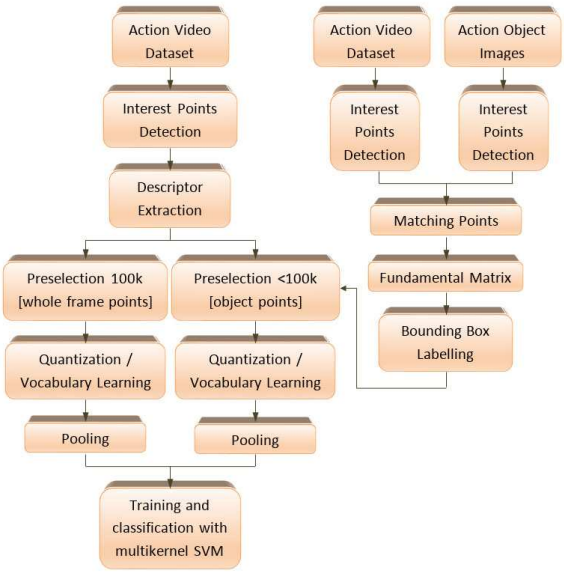


Figure 4.1: Scheme overview in the proposed approach.

encode the video frames computing a BoW description. Finally, we combine the two source of information, motion and context, by means of a multikernel SVM. Experimental results show that this procedure improves the recognition of actions.

4.3 Using Action Objects contextual Information

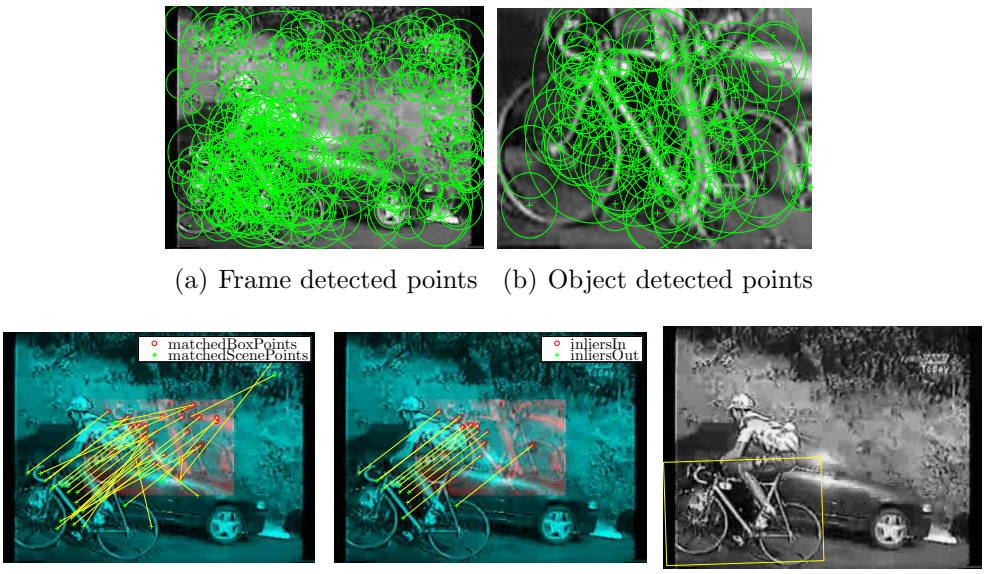
This approach is divided in two phases: first, we detect and track the object that is involved in the action performed, and second, we perform a BoW-based coding, with an information fusion by means of the multikernel SVM framework. An overview of the methodology is presented in Figure 4.1.

4.3.1 Object Detection and Tracking

In order to add contextual information related to the object appearing in an action, we must find the object in the video sequence. Each video contains one action, and we detect one object per action. Therefore, we obtain one instance image of each object per video and use this image to find the object along the whole video by matching a set of points previously extracted from the frame and the instance

image. The matching procedure based on the epipolar geometry described in Hartley and Zisserman (2004) is described in Figure 4.2. The points are extracted using both Harris corner detector and described by SURF features. This way ensures a large set of points belonging to the object, which is necessary to obtain good point correspondences and compute a representative bounding box. Then, we compute the point matching applying the k-Nearest Neighbor (kNN) algorithm and set a threshold to select the strongest matches.

Finally, we compute the fundamental matrix –excluding outliers by using Random Sample Consensus (Fischler and Bolles, 1981) – and use it to obtain a transformation of the initial bounding box. This ensures more accuracy around the area that limits the object in the frame. The result of this procedure is a bounding box enclosing the object used in each action for each frame in the video as can be seen in Figure 4.2.



(a) Frame detected points (b) Object detected points

(c) All matches (d) Inliers (e) Bounding box

Figure 4.2: First row: point detection and descriptor extraction for a video frame and the object image. Second row: matches and outlier filtering. Third row: transformed bounding box.

In order to add this information to the overall scheme, we first extract Spatio Temporal Interest Points (STIP) (Laptev, 2005) from each frame and video and compute their descriptor. Next, we select a maximum of 100k of object points applying the bounding boxes labels. Then, we construct a codebook from the pre-selected words belonging to objects and combine this codebook with others using

Table 4.1: Descriptors used to encode frames.

Descriptor	Characteristics	Reference
trajectories	KLT tracker or SIFT matcher	Jiang et al. (2012)
HOG	static appearance information from local gradients	Dalal and Triggs (2005)
HOF	local motion information	Lucas and Kanade (1981)
MBH	separately computes vertical and horizontal OF components	Dalal et al. (2006)
HOG3D	spatio-temporal extension of HOG	Kläser et al. (2008)

the multikernel SVM explained in the following section.

4.3.2 Multikernel for SVM

Visual features extracted from a video can represent a wide variety of information, such as scene (e.g., GIST (Solmaz et al., 2012)), motion (e.g., HOF (Lucas and Kanade, 1981), MBH (Dalal et al., 2006), HOG3D (Kläser et al., 2008)) or even just color (color histograms). To classify actions using all these features the information must be fused in an appropriate way. According to the moment of the combination, Snoek et al. (2005) proposed a classification of the fusion schemes in early or late fusion. In early fusion the descriptors are combined before training a classifier (e.g., concatenating (Ikizler-Cinbis and Sclaroff, 2010)), and in late fusion the classifiers are trained for each descriptor and the fusion is done for the results of all these classifiers (e.g., probabilistic fusion (Reddy and Shah, 2013)).

We use an early fusion in our approach since the combination is done before the training. A SVM with a chi-squared kernel is used for classification,

$$\chi^2(h_i, h_j) = \frac{1}{2}\sum_{k=1}^{n}\left(\frac{(h_i(k) - h_j(k))^2}{h_i(k) + h_j(k)}\right) \tag{4.1}$$

fusing all different descriptors by summing their kernel matrices normalized by the average distance.

$$K(h_i, h_j) = exp(-\sum_c \frac{1}{A^c}\chi^2(h_i^c, h_j^c)) \quad (4.2)$$

The value of A^c is the mean value of χ^2 distances between the training samples for the c-th channel (Zhang et al., 2006). All the weights are set to one, meaning that none of the kernels is more discriminative than the others.

4.4 Experimental Results and Discussion

Experiments are presented in this section. Firstly, we explain how to perform the setup, and secondly, the results are exposed in detail.

4.4.1 Experimental Setup

In this section object detection and tracking are considered in detail. Afterwards, we introduce the feature encoding in our evaluation step. Finally, the databases and their experimental setup are exposed.

4.4.1.1 Object Detection

The points used to identify and track the objects are a mixture of points obtained with Harris corner detector and features computed applying SURF. We use a threshold between 0,04 and 0,1 for Harris detector and a maximum number of 1000 points for SURF. This ensures enough quantity of points with enough quality belonging to the object, even in the case the object appearing in the video sequence is relatively small, like a ball or a sword. For the matching, we select the strongest 1% matches, which is restrictive but ensures better point correspondences.

4.4.1.2 BoW Based Encoding

To encode frames, we use the BoW approach. First, we make use of STIP points following the work of Laptev (2005). We compute different descriptors for each point, namely, HOG3D, trajectories, HOG, HOF, and MBH. In the case of HOG3D

descriptors, we set the parameters optimized for the KTH database as described in Kläser et al. (2008), resulting in 1008 dimensions in total. In the case of trajectories, HOG, HOF, and MBH, we follow the work of Wang et al. (2011) and set the parameters as they did. The dimensions of these descriptors are, respectively, 30 for trajectories, 96 for HOG, 108 for HOF and 192 for MBH, which are significantly smaller than HOG3D. We also compute DENSE_T, obtained as the concatenation of trajectories, HOG, HOF and MBH, which represents an early fusion and its dimension is 426. We train a codebook for each descriptor type using a maximum of 100k randomly sampled features. For the object kernel, we ensure the object point selection using the method described in Section 4.3.

Afterwards, we group the points employing the k-Means clustering algorithm with a maximum of 5 iterations. The size of the codebook is set to 500 words, following the works of Reddy and Shah (2013) and Bilinski and Corvee (2013) where the codebook size is limited to 500 or 1000 to avoid over-learning and despite the fact that the larger the number of clusters employed, the better the performance is. Finally, a SVM with an exponential chi-squared kernel is used for classification, combining all different descriptors by summing their kernel matrices and normalizing the result by the average distance. We use a 10 fold cross-validation with one-against-all approach. For all the experiments we employ the default parameter values in the LibSVM library (Chang and Lin, 2011).

4.4.1.3 Databases Used in the Experiments

As previously said, we test our approach with two different databases: KTH is collected by Schuldt et al. (2004) and does not contain any tool or object related to any action. Despite we can not take advantage of any contextual data, this experimentation allows us to test whether our approach is comparable to these of the state of the art. On the other hand, HMDB is collected by Kuehne et al. (2011) and is a more challenging and realistic one where objects used in actions are present.

The KTH database consists of 6 actions performed by 25 actors in a structured homogeneous environment with a total of 598 videos. The actions performed are

(a) Boxing (b) Hand waving (c) Hand clapping

(d) Running (e) Walking (f) Jogging

(g) Ride bike (h) Shoot gun (i) Shoot bow

(j) Draw sword (k) Swing baseball (l) Kick ball

Figure 4.3: Example frames from KTH database (first and second rows) and HMDB database (third and fourth rows). We use all the actions in KTH, that is, (a) boxing, (b) hand waving, (c) hand clapping, (d) running, (e) walking, (f) jogging, and a subset of the 51 actions in HMDB that include objects, (g) ride bike, (h) shoot gun, (i) shoot bow, (j) draw sword, (k) swing baseball, and (l) kick ball.

boxing, *hand-waving*, *hand-clapping*, *running*, *walking* and *jogging*, with no object involved in any of these actions. In order to reduce the computational burden, we pre-select 12 videos for any action performed by randomly selected actors into different environments, ensuring that as many variation as possible are employed, i.e., scene, person, illumination and camera distance, which makes a total of 72 videos.

The HMDB database consists of 51 actions from a total of 6,849 videos collected from a variety of sources ranging from digitized movies to YouTube videos.

Table 4.2: Comparison of different descriptors on the databases using our approach

Databases	**KTH** (%)	**HMDB** (%)	**HMDB + obj** (%)	**Δ** (%)
trajectories	45.51	38.13	39.81	1.68
HOG	70.63	54.29	64.76	10.47
HOF	62.99	41.67	44.78	3.11
MBH	61.55	38.3	47.10	8.8
DENSE_T	72.42	45.81	53.99	8.18
HOG3D	**86.57**	71.98	**79.58**	7.6

Considering that we need actions with object interaction, we do not follow the original splits proposed by Kuehne et al. (2011). We reduce the computational cost by pre-selecting 6 different actions with 20 videos per action, resulting in 120 videos in total. The pre-selected actions are *ride bike*, *shoot gun*, *shoot bow*, *draw sword*, *swing baseball* and *kick ball*. The purpose of this selection is dual: first, ensuring that an object is involved in the action, and second, ensuring the presence of as many variations as possible. Similar actions are also taken into account, a fact that makes the set more challenging.

Better and exhaustive explanation of the databases used and how they are split in affordable subsets is given in Appendix. A.

4.4.2 Experimental Results

We first analyze the use of multikernel SVM in Section 4.4.2.1. We want to know whether there is a difference in using a single kernel or a multiple kernel. Also, we compare the effect of different combinations of descriptors. In Section 4.4.2.2 we evaluate the impact of the addition of contextual information, based on the detection of the object related to the action.

4.4.2.1 Channel Selection

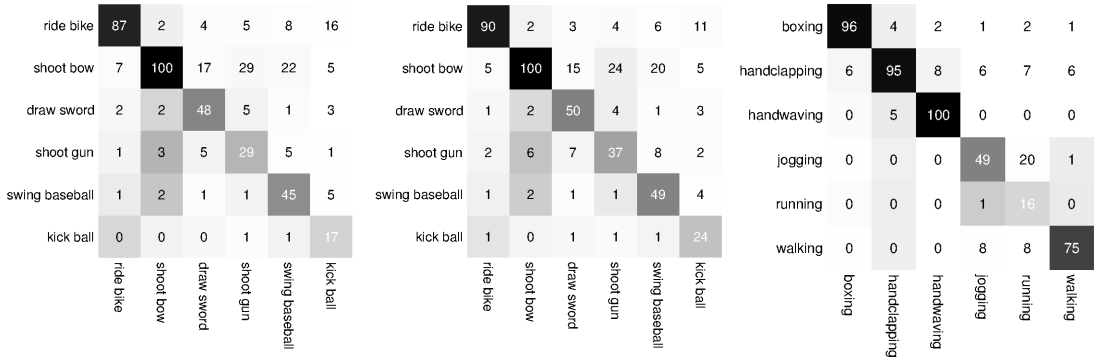
The use of a multikernel SVM allows us to add different descriptors into the traditional BoW approach for action recognition. This approach permits to include several descriptors into this scheme as explained in Wang et al. (2011), where a

Table 4.3: Comparison of different descriptors combinations on the databases with our approach

Databases	**KTH** (%)	**HMDB** (%)	**HMDB + obj** (%)	Δ (%)
trajectories + HOG	71.33	57.83	**71.57**	13.74
trajectories + HOF	62.77	43.00	48.64	5.64
trajectories + MBH	62.98	45.38	52.99	7.61
trajectories + HOG + HOF	79.83	64.67	69.39	4.72
trajectories + HOG + MBH	80.5	66.45	69.66	3.21
trajectories + HOF + MBH	74.68	53.44	57.45	4.01
trajectories + HOG + HOF + MBH	**82.94**	70.04	72.97	2.93
trajectories + HOG3D	73.67	61.56	64.07	2.51
HOG + HOF + MBH	81.66	68.09	70.23	2.14
HOG + MBH	77.13	60.39	66.92	6.53

combination of trajectories, HOG, HOF, and MBH is employed, and analyze how their combination by means of a multikernel SVM improves the performance with respect to any singular descriptor. In our work we do the same for a different set of descriptors, including trajectories, HOG, HOF, MBH, DENSE_T (an early fusion of them) and HOG3D. Results for all these descriptors using our approach can be seen in Table 4.2.

In our procedure, we have chosen a first descriptor and have progressively added new ones in order to see the effect of including new information into the kernel. To see the best improvements, we have chosen the descriptor that contributes the least, i.e., trajectories. These results can be seen in Table 4.3. Initially, this single descriptor gives a performance of 38.13%. Adding a descriptor that contributes more, HOG, the new value is 57.83%, which shows an improvement surpassing a 50% increase. On the other hand, adding another weak descriptor, HOF, the new value becomes 43.0%, which represents an improvement of a 12.8%. This fact shows the importance of choosing a good combination of descriptors. Almost all the additions improve the results, but the question is which one provides the best results since adding new channels results in higher computational costs. Therefore, we want the least number of channels that obtains the best results.



(a) HMDB (b) HMDB with objects (c) KTH

Figure 4.4: Confusion matrix for the (a) HMDB database using trajectories, HOG, HOF, MBH descriptors as it is done in Wang et al. (2011) with average performance for 500 codewords: 68.09%, (b) HMDB with our approach using the same configuration as (a), with average performance for 500 codewords: 70.23%, and (c) confusion matrix for the KTH database using trajectories, HOG, HOF, MBH descriptors as it is done in Wang et al. (2011). Average performance for 1000 codewords: 81.66%

4.4.2.2 Evaluation of Adding Contextual Information

In the case of the KTH database, where no objects are available, the present method equals the results of the multichannel approach in Wang et al. (2011). However, there is a significant improvement in the case where contextual information is present. In that case, our method outperforms the results obtained for all the descriptors, ranging from a minimum increase of 1.68% (HOF) to a maximum of 10.47% (HOG). The same happens when combination of descriptors are used and adding objects to the HOG + trajectories combination generates the highest increase, 13.74%, which also outperforms the rest of combinations. The highest value for each database is highlighted in boldface in Table 4.3.

The idea of contextual information influence can be seen in the confusion matrices in Figure 4.4. For example, *shoot bow* has confusions with the rest of actions, that is, 7% with *ride bike*, 17% with *draw sword*, 29% with *shoot gun*, 22% with *swing baseball*, and 5% with *kick ball*. After adding object information, these values are all reduced: 5% with *ride bike*, 15% with *draw sword*, 24% with *shoot gun*, 20% with *swing baseball* and 5% with *kick ball*, which means that the confusion of this action with respect to the rest is smaller as a consequence of including contextual information into the action description.

HOG3D and DENSE_T descriptors are used here to show two significant facts. First, that using a unique optimal descriptor is better than a combination of several descriptors that individually perform worse. This is apparent by the fact that HOG3D, which fuses information of space and time in a single descriptor, obtains a 71.98%. This result cannot be reached either by a concatenation of descriptors –trajectories, HOG, HOF, and MBH– or by a multikernel combination of the same descriptors, despite the latter is almost as good as HOG3D and reaches a performance of 70.04% while the former can at most get a value of 45.81%. Moreover, despite that no combination can outperform the best results reached by HOG3D, the addition of object information is able to increase the HOG3D result an extra 7.6%, up to 79.58%. Therefore, it is clearly stated that including contextual information always results in an improvement.

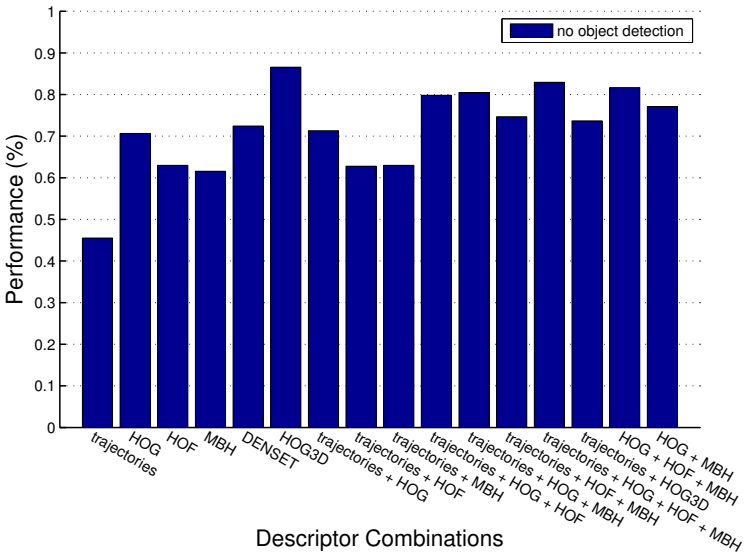


Figure 4.5: Evaluation of our approach for the KTH database.

Secondly, that combining descriptors is something that should be done with adequate criteria: Tables 4.2 and 4.3 show that early combination as a concatenation perform worse (45.81%) than using a late composition of trajectories, HOG, HOF, and MBH (70.04%) using a multikernel SVM. Figures 4.5 and 4.6 summarize all these results.

From the results obtained in this Section, we can state that there is a clear improvement in the action recognition task as a consequence of including contextual

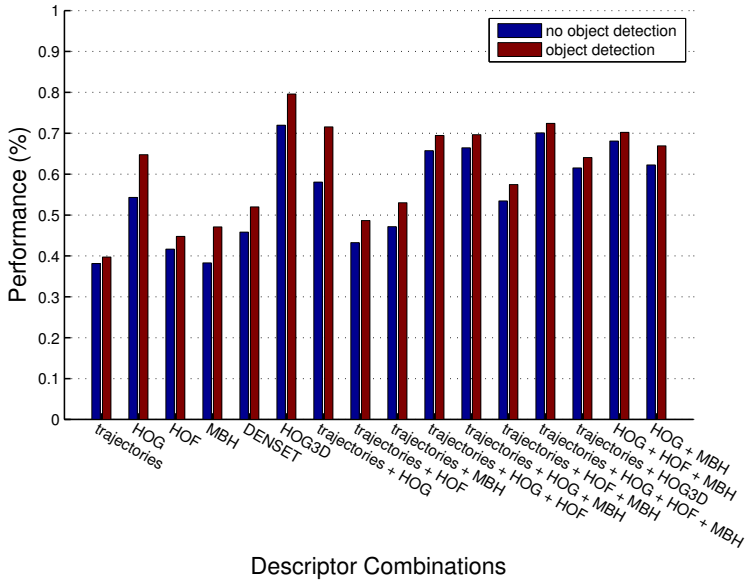


Figure 4.6: Evaluation of our approach using object detection for the HMDB database.

information in the action description and recognition. Moreover, the method presented in this chapter allows the obtaining and addition of such information.

4.5 Summary

We have proposed a method to incorporate action contextual information that extends a previous method used to combine motion related information into standard action recognition scheme based on BoW. This approach allows the addition of information related to the tool or object employ in the execution of an action and shows an increment of the overall recognition performance. We have shown that adding information without any specific purpose might lead to a lack of improvement adding the consequent computational cost to the scheme. Our approach complements space and time information and proposes a procedure to add any sort of contextual information that can be further generalized to include other data apart from the object used during an action. Additionally, the present approach shows that the best results are obtained when kernels from spacial, temporal, and tool information are combined into a multichannel SVM kernel. In this respect, the highest recognition rates are 71.57% using a combination of trajectories, HOG and object.

Chapter 5

Multimodal Sensoring

"The key to immortality is first living a life worth remembering."

- Bruce Lee,

5.1 Outline of the Chapter

Understanding human activities is one of the most challenging modern topics for robots. Either for imitation or anticipation, robots must recognize which action is performed by humans when they operate in a human environment. Action classification using a Bag of Words (BoW) representation has shown computational simplicity and good performance, but the increasing number of categories, including actions with high confusion, and the addition, especially in human robot interactions, of significant contextual and multimodal information has led most authors to focus their efforts on the combination of image descriptors. In this field, we propose the Contextual and Modal MultiKernel Learning Support Vector Machine (CMMKL-SVM). We introduce contextual information -objects directly related to the performed action by calculating the codebook from a set of points belonging to objects- and multimodal information -features from depth and 3D images resulting in a set of two extra modalities of information in addition to RGB images-. We code the action videos using a BoW representation with both contextual and modal information and introduce them to the optimal SVM kernel as a linear combination

of single kernels weighted by learning. Experiments have been carried out on two action databases, CAD120 and HMDB. The upturn achieved with this approach attained the same results for high constrained databases with respect to other similar approaches of the state of the art and it is much better as much realistic is the database, reaching a performance improvement of 14.27% for HMDB. The chapter is organized as follows:

- Section 5.2 introduces the problem and the related work existing on this topic.
- Section 5.3 presents the inclusion of sensor modes to the traditional BoW approach making use of contextual and multimodal information.
- Section 5.4 presents the experimentation and discussion.
- Section 5.5 summarizes the contribution of the approach.

5.2 Introduction

Analyzing video content has become critical in human robot interactions, where a robot must make a decision considering the information extracted from sensors such as cameras or lasers. In this context, our research focuses on the recognition of action in videos containing multimodal and contextual information about the means by which an action is carried out. Some public databases are conformed by a set of RGB videos where scenes and parameters such as illumination, focus, distance, and viewpoints are mostly controlled, and few information exits about the tools and objects that were involved in the action. In robotic contexts, it is usual to have multimodal information, provided by distance laser sensors or by 3D cameras such as Kinect.

CAD120 database (Koppula et al., 2013) is recorded with a high controlled environment, which is ideal for human-robot interactions, although it includes both contextual and multimodal information. This database contains 10 high level actions performed by 4 different subjects which in total corresponds to 124 manually annotated videos. However, in order to go beyond the current state of the art in action recognition topic for real videos, more realistic databases have been

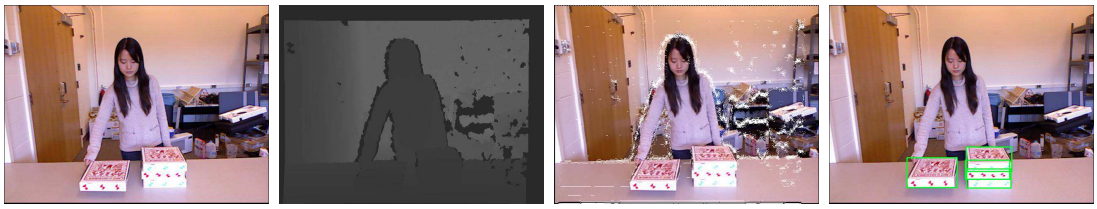


Figure 5.1: Multimodal database CAD120 with RGB (most left), Depth map (middle left), 3D map (middle right), object context (most right).

increasingly employed, including videos that stage more realistic actions.

HMDB Kuehne et al. (2011), is one of the largest action video database to-date with 51 action categories, which in total contains 6849 manually annotated clips extracted from a variety of sources ranging from digitized movies to YouTube videos. This database has been created to evaluate the performance of computer vision systems for action recognition and explore the robustness of these methods under various conditions such as cluttered backgrounds, fast irregular motions, occlusions and camera motion. In this database, actions with contextually connected objects can be found, although no multimodal recording is available.

Specifically, we select from the HMDB database a subset of actions that are performed employing a tool or object. This contextual information allows the computer to discriminate apparently similar actions such as the case of shooting a gun or a bow. The biggest difference among these similar actions lies in the tool employed to carry out the action.

In this chapter, we detail how these different sources of information -depth, objects- can be combined in a richer description of human actions that permits higher recognition rates. In order to increase the robustness of the recognition of actions in more challenging situations, we propose to weight different sources of information relevant to discriminate actions, namely, the spatio-temporal features that describe motion by RGB and depth modes, and the contextual information that explains how an action is carried out by object features. The Figure 5.1 shows sample images from CAD120 database, representing the same frame of a video as four different sources of information: RGB, depth and precomputed 3D images and the objects detected in this frame.

The main contribution of the work in this chapter is the fusion and discrimination of new information sources for performed actions with a recognition structure that weights the addition of new information using a multichannel SVM. The use of the multichannel SVM has previously proven very successful in action recognition (Wang et al., 2013; Bilinski and Corvee, 2013). Thus, we take advantage of this structure in two ways: firstly, by adding data that is strictly not a descriptor of motion but modal or contextual information obtained by segmenting the region where the action takes place in three space dimensions and describing the tool employed in the action. Secondly, by weighting the channels with a multikernel learning approach, determining which channel has more non-redundant information.

Related Work

Local space-time features (Laptev, 2005) have been shown to be successful for general action recognition because they avoid non-trivial pre-processing steps, such as tracking and segmentation, and provide descriptors invariant to illumination and camera motion. In particular, HOG3D (Kläser et al., 2008) has proven to outperform most of the descriptors of the same kind.

Experimenting in robotic environments, contextual and multimodal information have been considered in action recognition frameworks. Works of Pieropan et al. (2014) and Tsai et al. (2013) fuse information into two different stages with respect to the training, that is, before and after it respectively. Snoek et al. (2005) studied the different methods of descriptor fusion and classified them into early fusion and late fusion approaches. The former consists of a fusion before the training step, while the latter is a fusion afterwards. In this context, Ikizler-Cinbis and Sclaroff (2010) combined six different visual descriptors for three different contextual information types, namely, *people* (HOF and HOG3D), *objects* (HOF and HOG), and *scene* (GIST and color histograms) by using a multiple MIL approach, which is a concatenation of bag representations and classified with an L2-Regularized Linear SVM. In the work of Bilinski and Corvee (2013), a multichannel χ^2 kernel SVM is

used for the combination of a set of descriptors. Similarly, the work of Wang et al. (2013) computes dense trajectories and their descriptors to finally combine them using an averaged multichannel SVM.

Considering Multiple Kernel Learning (MKL) as an early fusion approach, it was first proposed in Lanckriet et al. (2004). SVM approaches focus their efforts on how to improve the classification accuracy by exploiting different formulations and how to improve learning efficiency by exploiting different optimization techniques. Bucak et al. (2014) showed that conflicting statements exist which are largely due to the variations in the experimental conditions. In the work it is also stated that while some studies reported that averaging kernels (same weight for each kernel) is outperformed by SVM (Bucak et al., 2010), others conclude the opposite (Gehler and Nowozin, 2009). Linear combinations do not have to deal with non-convex optimization problems which would lead them to poor computational efficiency and suboptimal performance. That is the reason why most of the authors prefer them instead of non linear combinations.

Traditional kernel combination learning approaches based on the SVM wrapper SimpleSVM (Rakotomamonjy et al., 2008) are mainly focused on the usage of the same training data, making use of linear, polynomial or RBF kernels. This fact is in contrast to recently published works on the multichannel approach in Wang et al. (2011) and later Wang and Schmid (2013), which combined different training data by kernel average.

In our work, unlike the aforementioned state-of-the-art methods, we consider depth, 3D information, and image descriptors of the objects used in the actions by means of a BoW-based action recognition approach. To this end, we first detect the set of points belonging to the object as explained previously in Chapter 4. Then, we compute codebooks for each video mode and context descriptors. Finally, we combine the three sources of information, motion, depth and objects, by weighting a multikernel SVM using CMMKL-SVM. Experimental results show that this procedure improves the recognition rate of actions.

5.3 Combining Contextual and Modal Action Information into a Weighted Multikernel SVM

5.3.1 RGB, Depth and 3D Multimodal

RGB images are usually provided by a single camera mounted in the body of the robot or in a fixed place in the space. That imposes the limitation of a single view of the performed action. There exist databases which consider the possibility of a multiple viewpoint, introducing more variability to the information captured. That would be the case if different robots were analyzing the same action simultaneously in different positions, but we consider human-robot interactions that involve just one robot. Hence, we test our algorithm over a database which provides depth maps, i.e. CAD120.

We make use of depth information in two ways: first, extracting descriptors as done with the RGB video sequences. We have, then, a set of descriptors such as trajectories, HOG, HOF, MBH for RGB and Depth. Depth sequences allow to differentiate elements in the scene like background and objects over planes different from the one in which the action takes place. Second, generating a RGB-D sequence in which we can extract 3D spatial descriptors, such as FPFH. 3D sequences provide 3D spatial information combined in one descriptor. In the end, RGB, Depth and 3D descriptors generate independent codebooks.

5.3.2 Object Detection and Tracking as Context

Following the work done in Chapter 4 and considering that each video contains one action, we detect the objects that are employed in the performance of this action. We make use of the matching procedure based on the epipolar geometry, that computes the Fundamental Matrix between two consecutive frames and extracts the bounding boxes for each object in each frame. The result of this procedure is a set

of bounding boxes that enclose the objects used in each action for each frame in the video ensuring high accuracy around the area that limits the objects. We also limit the computational burden by keeping a maximum of 100k points belonging to objects applying bounding box labels when creating the codebook. What contrasts with the procedure described in Chapter 4 is that in this case more that one object is involved when performing the action. Thus, all these objects are consequently detected and tracked.

5.3.3 CMMKL-SVM

Visual features extracted from a RGB video can represent a wide variety of information, such as scene (e.g., GIST (Solmaz et al., 2012)), motion (e.g., HOF (Lucas and Kanade, 1981), MBH (Dalal et al., 2006), HOG3D (Kläser et al., 2008)) or even just color (color histograms). In our approach we include extra features, such as depth and 3D scene information (e.g. FPFH (Rusu, 2009)), and object related information. To classify actions using all these features, the information must be fused in an appropriate way. According to the moment of the combination, Snoek et al. (2005) proposed a classification of the fusion schemes in early or late fusion. Multikernel approaches use early fusion since the combination is done before the training.

The work of Wang et al. (2013) use a linear combination of different kernels, calculated from a set of codebooks generated with different descriptors. A SVM with a χ^2 kernel for classification is used,

$$\chi^2(h_i, h_j) = \frac{1}{2}\sum_{k=1}^{n}\left(\frac{(h_i(k) - h_j(k))^2}{h_i(k) + h_j(k)}\right) \tag{5.1}$$

ensuring that the kernel matrices are strictly positive definite. They fused different descriptors by summing up the corresponding kernel matrices, normalized by the average distance A^c of χ^2 distances between the training samples for the c-th channel. No kernel weighting is done, so no kernel is more discriminative than the others.

In our approach, given the base kernels

$$K_c(h_i, h_j) = exp(-\frac{1}{A^c}\chi^2(h_i^c, h_j^c)) \tag{5.2}$$

the optimal kernel of a certain descriptor is approximated as

$$K_{opt} = \sum_c d_c K_c \tag{5.3}$$

where d_c is the kernel weight for c-th channel. Each K_c represents the precoded c-th information referred to the action.

The optimization is carried out within a SVM framework that achieves the best classification on the training set subject to a regularization scheme. In this formulation, the objective function is near identical to the standard L_1 C-SVM objective function. The regularization prevents the weights from becoming too large, although this could be achieved by requiring that the weights sum up to the unit but also restricting the search space.

$$\begin{aligned} \underset{w,d,\xi}{\text{minimize}} \quad & \frac{1}{2}w^t w + C1^t\xi + \sigma^t d \\ \text{subject to} \quad & y_i(w^t K_c + b) \geq 1 - \xi_i \\ & \xi \geq 0, d \geq 0, Ad \geq p \end{aligned} \tag{5.4}$$

The constraints are also similar to the standard SVM formulation, with the addition of two constraints. First, $d \geq 0$, which ensures that the weights can be interpreted and also leads to a much more efficient optimization problem. Second, $Ad \geq p$, with some restrictions, that allow us to encode prior knowledge about the problem.

In order to tackle large scale problems involving hundreds of kernels, we adopt the minimax optimization strategy and solve the problem by using projected gradient descent, taking care to ensure that the constraints $dn + 1 \geq 0$ and $Adn + 1 \geq p$ are satisfied. This algorithm proceeds in two stages. In the first one, weights d_c are maximized and Support Vectors (SV) are obtained. In the second stage, objective function is minimized by projected gradient descent. The two stages are repeated

until convergence or a maximum of the number of iterations is reached, at which point the weights d and $SV's$ are obtained.

5.4 Experimental Results and Discussion

Experiments are presented in this section. Firstly, we explain how to perform the setup, and secondly, the results are exposed in detail.

5.4.1 Experimental Results

In this section, modal selection and object detection and tracking are considered in detail. Afterwards, we introduce the encoding framework based on BoW. Finally, the databases and their experimental setups are exposed.

5.4.1.1 Extracting Contextual Information: Objects

The points used to identify and track the objects are a mixture of RGB points obtained using Harris corner detector and features computed applying SURF. We use a threshold between 0,04 and 0,1 for Harris detector and a maximum number of 1000 points for SURF. This ensures enough quantity of points with enough quality belonging to the object, even in the case that the object appearing in the video sequence is relatively small, like a ball or a sword. For the matching, we select the strongest 1% of matches, which is restrictive but ensures better point correspondences. These considerations refer mainly to HMDB database, which is more realistic than CAD120. Object detection and tracking for CAD120 are more accurate due to their highly controlled conditions.

5.4.1.2 Extracting Multimodal Information: RGB, Depth and 3D

We select three informational modes taking advantage of the RGB-D videos, forming the set with RGB, depth and 3D videos.

First, for each point in RGB and Depth videos we compute different descriptors, HOG3D, trajectories, HOG, HOF, MBH. In the case of HOG3D descriptors, we set

the parameters optimized for KTH database as described in Kläser et al. (2008), which have demonstrated a good performance not only for the KTH set, resulting in 1008 dimensions in total. In the case of trajectories, HOG, HOF and MBH, we follow the work of Wang et al. (2013) and set the parameters likewise. The dimensions of these descriptors are, respectively, 30 (trajectories), 96 (HOG), 108 (HOF) and 192 (MBH), which are significantly smaller than these of HOG3D. We set same parameter values for both, RGB and Depth videos.

Second, we consider the FPFH descriptor (Rusu, 2009) of the 3D Point Cloud Library. We configure the descriptor length to FPFHSignature33, that creates a 33 dimension descriptor. We set the FPFH radius search to 100 in order to ensure enough valid descriptors.

5.4.1.3 Encoding Using BoW

We use the BoW approach to encode frames. First, we make use of STIP points following the work of Laptev (2005). We compute different descriptors for each point in RGB videos, Depth videos and 3D videos. We train a codebook for each descriptor type using a maximum of 100k randomly sampled features. For the object kernel, we ensure the object point selection using the method described in Chapter 4.

Afterwards, we group the points employing the k-Means clustering algorithm with a maximum of 5 iterations which ensures enough convergence. In order to compare results with the ones obtained in Chapter 4, the size of the codebook is set to 500 words, avoiding over-learning, despite the fact that the larger the number of clusters employed, the better the performance is. Finally, a SVM with an exponential χ^2 kernel is used for classification, using a 10 fold cross-validation method with the one-against-all approach. For all the experiments we employ the default parameter values in the LibSVM library (Chang and Lin, 2011).

5.4.1.4 Multikernel Selection

We perform a CMMKL-SVM for classification that uses the default parameters in Vedaldi et al. (2009). We precalculate each kernel based on image coders (objects, 3D, Depth, RGB descriptors) and perform a train in order to obtain the best combination of weights.

In the comparison step, we also perform a uniformly weighted combination by summing their kernel matrices and normalizing the result by the average distance as in Wang et al. (2013).

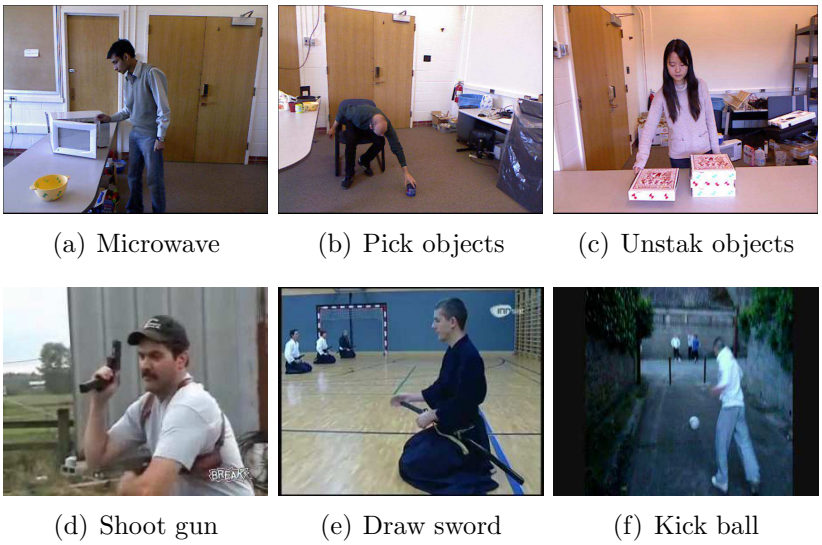
5.4.1.5 Databases

We test our model with two different databases, CAD120 (Koppula et al., 2013) and HMDB (Kuehne et al., 2011). CAD120 contains objects that involve actions in a highly controlled environment and multimodal information such as RGB and depth videos. HMDB is a more challenging and realistic one, where objects used in actions are present. Although no 3D information exists, we use this dataset to test our approach and compare the results to the state-of-the-art results. Sample frames for each database are shown in Figure 5.2, in which three actions from the whole collection are represented for both databases.

The CAD120 database contains 124 RGB-D videos of 4 different subjects performing 10 high-level actions. Each action is performed three times with different objects. It contains a total of 61585 3D video frames. The actions have a long sequence of subactivities which might be considered in future work.

The 10 high-level actions performed are *arranging objects*, *cleaning objects*, *having meal*, *making cereal*, *microwaving food*, *picking objects*, *stacking objects*, *taking food*, *taking medicine* and *unstaking objects*.

The HMDB database consists of 51 actions from a total of 6,849 videos collected from a variety of sources ranging from digitized movies to YouTube videos. The action categories are grouped in five types: general facial actions, facial actions with object manipulation, general body movements, body movements with object interaction, and body movements for human interaction.



(a) Microwave (b) Pick objects (c) Unstak objects

(d) Shoot gun (e) Draw sword (f) Kick ball

Figure 5.2: First row: example frames from the CAD120 database showing three out of ten actions, (a) microwaving food, (b) picking objects, (c) unstacking objects. Second row: example frames from three actions from the subset selected of the 51 actions in HMDB which include objects, (d) shoot gun, (e) draw sword, (f) kick ball.

In order to obtain comparable results and considering that we need actions where an object is used, we do not follow the original splits proposed by Kuehne et al. (2011). Instead, we use the split detailed in Appendix A, which ensures the presence of as many variation as possible by following a proportion of clips similar to that in the complete database. These variations include what part of the body is shown, the number of people involved in the action, the camera motion and viewpoint, and the quality of the video.

The split consists of 6 different actions with 20 videos per action, resulting in 120 videos in total. These actions are *ride bike*, *shoot gun*, *shoot bow*, *draw sword*, *swing baseball* and *kick ball*. The purpose of this selection is dual: first, ensuring that an object always appears in the action, and second, ensuring the presence of as many variations as possible. Similar actions like *draw sword* and *swing baseball* are also taken into account, a fact that makes the set more challenging.

5.4.2 Evaluation and Discussion

In Section 5.4.2.1 we evaluate our CMMKL-SVM approach on CAD120 and HMDB datasets. We first evaluate single descriptors in order to find the most significant

Table 5.1: Comparison of different descriptors on the databases

Databases	**CAD120** (%)	**HMDB** (%)
RGB trajectories	32.30	38.13
RGB HOG	70.17	54.29
RGB HOF	49.02	41.6
RGB MBH	46.97	38.30
RGB HOG3D	**83.94**	**71.98**
Depth trajectories	56.34	n/a
Depth HOG	55.99	n/a
Depth HOF	56.47	n/a
Depth MBH	55.18	n/a
FPFH	60.51	n/a

ones. Later, we evaluate the combination of different kernels and obtain the weights that informs us of the relevance of each kernel. In section 5.4.2.2 we compare our results for each database with those of the state of the art.

5.4.2.1 Evaluating CMMKL-SVM

The use of CMMKL-SVM allows us to add different descriptors into the standard BoW approach for action recognition. This approach permits the inclusion of several image descriptors into this scheme as explained in Chapter 4, and reduces the effect of information redundancy weighting a multikernel SVM. This approach improves the performance with respect to any singular descriptor or an averaged combination of them.

In our first experiment we calculate the average accuracy for each of the following descriptors: trajectories, HOG, HOF, MBH, HOG3D, Depth_trajectories, Depth_HOG, Depth_HOF, Depth_MBH, Depth_HOG3D and FPFH. As we can see in Table 5.1, HOG3D descriptor gives the best action recognition performance. HOG3D avoid non-trivial pre-processing steps, such as tracking and segmentation, fuses 2D space and time information, and provides descriptors invariant to illumination and camera motion. This aspect shows that using a unique optimal descriptor can be better than a combination of several descriptors that perform worse individually.

Table 5.2: Context and modal influence on the databases using two approaches: ours (CMMKL) and Uniformly Weighted (UW) likewise Chapter 4

CAD120 Database	**UW (%)**	**CMMKL (%)**	**Kernel Weights**
Object info. combined with RGB descriptor			
obj+RGB traj.	36.34	56.57	0.5/0.7
obj+RGB HOG	75.21	86.32	0.2/0.8
obj+RGB HOF	54.78	74.94	0.4/0.8
obj+RGB MBH	54.50	68.32	0.4/0.6
Depth info. combined with RGB descriptor			
Depth+RGB traj.	72.61	83.19	0.7/0.3
Depth+RGB HOG	81.63	89.59	0.9/0.9
Depth+RGB HOF	75.08	86.53	0.4/0.8
Depth+RGB MBH	79.03	87.96	0.1/0.4
3D info. combined with RGB descriptor			
FPFH+RGB traj.	62.03	87.98	0.6/0.4
FPFH+RGB HOG	69.98	90.67	0.2/0.5
FPFH+RGB HOF	67.65	89.28	0.7/0.7
FPFH+RGB MBH	69.27	90.18	0.8/0.7
HMDB Databases	**UW (%)**	**CMMKL (%)**	**Kernel Weights**
Object info. combined with RGB descriptor			
obj+RGB traj.	39.81	55.43	0.1/0.2
obj+RGB HOG	64.76	86.72	0.5/0.7
obj+RGB HOF	44.78	65.37	0.3/0.3
obj+RGB MBH	47.10	61.61	0.8/0.5

This is apparent in the fact that HOG3D obtains a 71.98% for HMDB and 83.94% for CAD120.

An extra objective of our approach is to overpass this performance by using CMMKL-SVM with the best weighted combination of descriptors using RGB videos, Depth videos, 3D points and objects. That would considerably reduce the time of the overall procedure, taking into account that HOG3D is quite computationally expensive. Additionally, Depth descriptors give similar results, (55%), for each single descriptor -trajectories, HOG, HOF and MBH- meaning that these descriptors lose their singular characteristics when used for depth videos. On the other hand, HOG and FPFH are the best choices when used as single descriptors, obtaining a

recognition rate of 70.17% and 60.51% respectively in CAD120. This is due to the fact that they give spatial information of the action, a fact that has been verified in Chapter 4.

In the second experiment, our purpose is to observe the influence of the context (objects) and modes (Depth, 3D) when employing single descriptors (trajectories, HOG, HOF, MBH) on RGB videos. The results are shown in Table 5.2. We perform the experiments with our approach CMMKL-SVM and the uniformly weighted approach from Chapter 4. We show that the fusion of context and mode information in a MKL framework is better than averaging kernels. Having a look at the results in Table 5.2 we can observe that the addition of context, gives an important improvement of 20% on the average recognition rate for every trial in HMDB when using our approach. For CAD120, this improvement is much lower than for HMDB, 10% on average, due to the quality of the videos and the lack of extensive variability in conditions such as illumination and viewpoint. In Table 5.2 we show how context, depth or 3D information always outperforms the recognition accuracy reached using a single descriptor.

The third experiment wants to find the best combination between all of the descriptors. The experiment has been performed choosing a first descriptor and progressively adding new ones in order to see the effect of the inclusion of this new information into the CMMKL-SVM. To see the best improvements, we have chosen the descriptor that contributes the least, i.e., trajectories. These results can be seen in Table 5.3. Any additional information improves the results, but the question is which one provides the best results since adding new channels results in higher computational costs. Therefore, we want the least number of channels that provides the best results. We can conclude from these results that the addition of descriptors which provides redundant information leads to a lack of improvement. For example, the addition of HOF, object or FPFH to the combination trajectories + HOG leads to no significant improvement. We must observe that HOF provides temporal information in a similar sense as trajectories.

Table 5.3: Using different descriptors combinations on the databases with our approach

Database	**CAD120**		
	UW (%)	**CMMKL (%)**	**Kernel Weights**
trajectories+HOG	71.04	83.24	0.2/0.6
trajectories+HOF	49.94	71.1	1.0/0.7
trajectories+Depth_HOG	73.28	79.15	0.10/0.81
trajectories+HOG+HOF	75.10	85.94	0.5/0.5/0.1
trajectories+HOG+obj	71.40	83.60	0.8/0.9/0.4
trajectories+HOG+FPFH	71.90	90.33	0.8/0.4/0.6
trajectories+HOG+HOF+MBH	77.71	87.60	0.1/0.2/0.3/0.4
trajectories+HOG+HOF+obj	75.30	84.66	0.9/0.5/0.5/0.7
trajectories+HOG+HOF+Depth_HOG	**84.78**	90.70	0.9/0.7/0.4/0.9
trajectories+HOG+HOF+FPFH	76.04	89.75	0.8/0.6/0.4/0.2
trajectories+HOG+HOF+MBH+obj	77.92	85.21	0.3/1.0/0.5/0.8/0.1
trajectories+HOG+obj+FPFH+Depth_HOG	79.73	**92.83**	0.0/0.6/1.0/0.8/0.5
Database	**HMDB**		
	UW (%)	**CMMKL (%)**	**Kernel Weights**
trajectories+HOG	57.83	82.24	0.3/0.4
trajectories+HOF	48.64	65.28	0.3/0.3
trajectories+HOG+HOF	64.67	85.82	0.1/0.8/0.3
trajectories+HOG+obj	**71.57**	**85.84**	0.3/0.7/0.1
trajectories+HOG+HOF+MBH	70.04	80.69	0.7/0.8/0.1/0.5
trajectories+HOG+HOF+obj	69.39	85.36	0.2/0.9/0.7/0.6
trajectories+HOG+HOF+MBH+obj	72.97	85.41	0.6/0.1/0.4/0.0/0.2

5.4.2.2 Discussion

Comparing to the state-of-the-art, on one hand, Koppula et al. (2013) obtained a 93,5% in CAD120 database using a CRF-based approach. We obtain a similar recognition accuracy of 92.83% using CMMKL-SVM. On the other hand, we significantly improve the results for HMDB, where we used in Chapter 4 a fusion of objects and RGB descriptors by averaging a multikernel SVM reaching 71,57%, much lower than the present score of 85.41%. Table 5.4 shows this comparison for CAD120 and Table 5.5 for HMDB. The more realistic the database is the more relevant the acquisition and weights of contextual and multimodal information are.

Referring to Table 5.3, we can see the importance of weighting channels. Using a kernel averaging scheme likewise we did in Chapter 4 we always obtained a lower

performance than this new approach, which takes into account the redundancy of information introduced by similar descriptors. This can be seen in the combination trajectories + HOG + HOF, where trajectories almost loses its importance (0.1) because of other descriptors such as HOF (0.3), which also provides temporal information like trajectories. However, HOG still remains the most significant descriptor (0.8). This reinforces the hypothesis made in Chapter 4 that the strongest descriptors are those that provide spatial information.

Finally, regarding the confusion of the actions, CMMKL-SVM reduces confusion between actions, even for similar actions, as can be seen in Figure 5.3. For example, *Unstacking objects* for CAD120 is easily confused with *Stacking objects*, a relation that averaging kernels cannot break (1%) but our approach does (32%). The same happens when *kicking a ball*, where averaging kernels performs a 24% and CMMKL a 44%. In general, all actions in both databases have their confusion index reduced. Therefore, the overall performance of our action recognition approach is higher than other state-of-the-art approaches.

Table 5.4: Comparison to the state of the art on CAD120 database

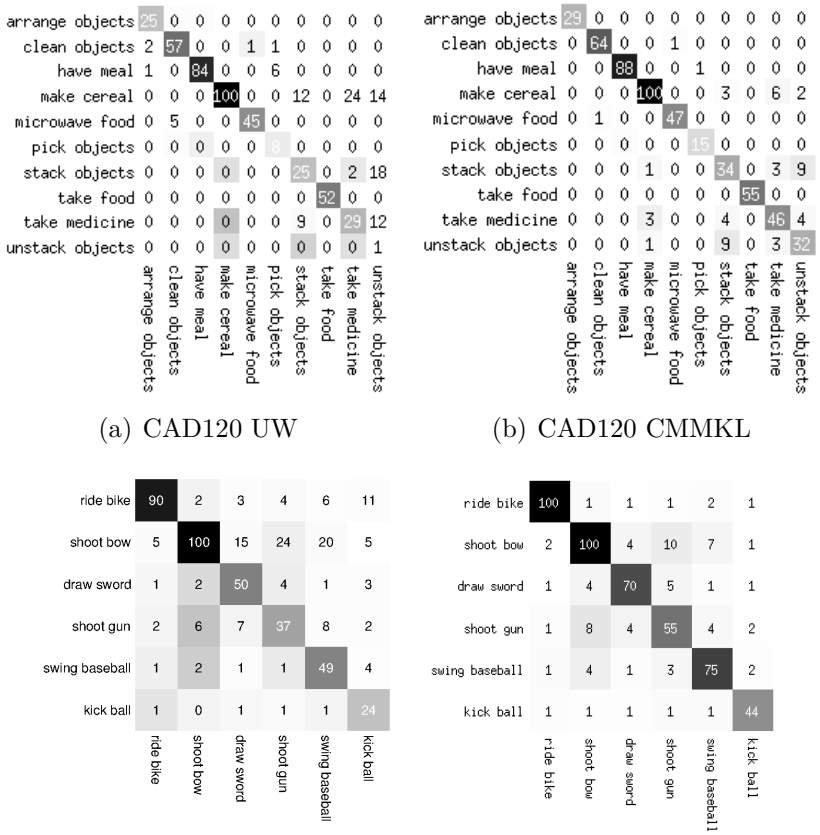
Work	Approach	Avg. acc.
Koppula et al. (2013)	CRF-based	93.50 %
Ours	CMMKL-SVM	**92.83** %

Table 5.5: Comparison to the state of the art on HMDB database

Work	Approach	Avg. acc.
Chapter 4	Multichannel UW	71.57 %
Ours	CMMKL-SVM	**85.84** %

5.5 Summary

In this paper we have proposed a methodology to combine different descriptors within a standard action recognition scheme based on BoW. Our approach adds information related to the objects, depth maps and 3D points, and shows an increment of the overall action recognition performance. The addition of the extra



(a) CAD120 UW (b) CAD120 CMMKL

(c) HMDB UW (d) HMDB CMMKL

Figure 5.3: Confusion matrices for: (a) CAD120 database using objects, FPFH, Depth HOG, trajectories, HOG using UW approach (Chapter 4) with average performance for 500 codewords: 79.73%, (b) CAD120 with our approach using the same configuration as (a), with average performance for 500 codewords: 90.83% (c) HMDB database using objects, trajectories, HOG, HOF, MBH descriptors as it is done in Chapter 4 with average performance for 500 codewords: 72.97%, (d) HMDB with our approach using the same configuration as (a), with average performance for 500 codewords: 85.41%.

image descriptors, either from RGB context or sensor modality, leads to an increment of the computational cost. As a consequence, it is important to discriminate, or even discard, the less important descriptors. Our approach complements space and time information extracted with video descriptors, and proposes a procedure to incorporate and weight any contextual and modal information that can be further generalized to include other data provided by new context descriptors and/or new devices. Additionally, the present approach also shows that the best results are obtained when kernels from spatial, temporal, context, 3D points and depth are combined within the CMMKL-SVM approach. In this respect, the highest recognition rates (92.83%) have been obtained when a combination of trajectories,

HOG, FPFH, Depth and object is used. Due to the relevant importance to intelligent robots, our future work will focus on the improvement of multimodal fusion and the reduction of the computational burden by exploiting different optimization techniques for MKL, allowing a quicker response of the robot to interact with humans by either imitating or anticipating actions.

Chapter 6

Incremental Learning

"There is a difference between knowing the path and walking the path."

- Andy and Larry Wachowski, *The Matrix*

6.1 Outline of the Chapter

This chapter presents an Incremental Weighted Contextual and Modal MultiKerneL Support Vector Machine (IWCMMKL-SVM) approach for improving human action recognition. Different frame coding is performed based on multiple information sources, namely, RGB and Depth videos, 3D Videos, and context. This approach allows for the incorporation of the new action demonstrations without performing a new training from batch. During the incremental step, new frames are coded likewise the previous training, and the action descriptors are merged with the Support Vectors (SV) that characterize the old SVM classifier. The proposed approach is evaluated over two datasets, HMDB and CAD120. The results indicate that although the incremental procedure reduces the amount of information used for classification compared to the batch learning method, the overall performance is at least maintained thanks to the discriminatory capacity of the weighted support vectors. The chapter is organized as follows:

- Section 6.2 introduces the problem and the related work existing on this topic.

- Section 6.3 presents the incremental approach proposed based on the Support Vectors of the SVM classifier.
- Section 6.4 presents the experimentation and discussion.
- Section 6.5 summarizes the contribution of the approach.

6.2 Introduction

Video analysis has become critical in human robot interactions, where a robot must make a decision considering the information extracted from robot joint sensors, accelerometers, cameras or lasers. In this context, our research is focused on the recognition of action in videos containing multimodal and contextual information. Furthermore, in this chapter we extend the traditional BoW approach to an incremental one, so that new data can be incorporated in the trained model. This feature is applicable to different fields of research, as for example in imitation learning, where the approach allows the robot to learn from its own action performance when performing an imitation.

In order to test this new approach we focused our efforts on experimenting and making use of well known recorded and collected databases. Public databases are usually conformed by a set of RGB videos where scenes and parameters such as illumination, focus, distance, and viewpoints are mostly controlled, and little information exists about the tools and objects that were involved in the action. Furthermore, as the need for sophisticated gathered data increases in some fields of research, e.g. in robotic environments, multimodal information, provided by distance laser sensors or by 3D cameras such as Kinect, is incorporated in more recent databases.

One of the most complete databases is CAD120 (Koppula et al., 2013). It is recorded in a highly controlled environment, which is ideal for human-robot interactions and it includes both contextual and multimodal information. This database contains 10 high level actions performed by 4 different subjects which in total corresponds to 124 manually annotated videos.

However, in order to go beyond the current state of the art in the action recognition topic for real videos, more realistic databases have increasingly been employed, including videos that stage more realistic actions. HMDB (Kuehne et al., 2011), is one of the largest action video databases to-date with 51 action categories, which in total contains 6766 manually annotated clips extracted from a variety of sources ranging from digitized movies to YouTube. This database has been created to evaluate the performance of computer vision systems for action recognition and explore the robustness of these methods under various conditions such as cluttered backgrounds, fast irregular motions, occlusions and camera motion.

In this chapter, the different sources of information present in the databases -depth, 3D and objects- are combined in a richer description of human actions that permits higher recognition rates. In order to increase the robustness of the recognition of actions in more challenging situations, we weight different sources of information relevant to discriminate actions likewise we did in Chapter 5, namely, the spatio-temporal features that describe motion by RGB, depth and 3D modes, and the contextual information that explains how an action is carried out by object features. Finally, the most interesting contribution is that the work of this chapter proposes an incremental approach that allows the incorporation of new training data to the classifier without having to train it again from batch.

Related Work

Despite the intensive work recently done with other learning algorithms, such as deep learning, SVM remains as one of the most widely used kernel learning algorithm, and it has been successfully used for many problems. The generalization property of an SVM depends on its Support Vectors (SV), which is a good enough description of the decision bound. Unfortunately, the traditional SVM approach is very time and space consuming due to that the training is usually a Quadratic Programming (QP) problem, a fact that makes it unaffordable for large-scale problems. SVM can actually be improved in these two targets: first, by realizing online incremental learning, which

is useful especially when all the data cannot be loaded into the memory at the same time or under the online condition; second, by solving large-scale problem, which is important because the real-world data are often large-scale data.

Furthermore, being able to adapt learning methods to online or incremental learning is an important feature that makes them feasible for real world problems. Some reasons to adopt incremental learning are: firstly, incremental learning may be used to keep the memory and computing consumption at an accomplishable level; secondly, incremental learning is appropriate when the useful training examples cannot be collected before the learning process begins, for example the stream data. Zhou and Chen (2002) classified incremental learning in three methods: example, class and attribute incremental learning, which incorporate new examples, new classes, and new attributes to the trained learning model respectively.

To evaluate the effectiveness of incremental training, Syed et al. (1999) proposed an Incremental Support Vector Machine (ISVM), which is considered as one of the first SVMs with incremental learning. However the work gives only approximate results and has been then extended and developed, such as with the SV-L-incremental algorithm (Rüping, 2001) and NORMA algorithm (Kivinen et al., 2004). Similarly, Hai et al. (2010) exposed an incremental learning algorithm for SVM based on the voting principle, and Wu et al. (2008) made use of the convex hulls algorithm in order to reduce computational cost for SVM incremental learning, defining a new approach called Convex Hull Support Vector Machine (HC-SVM).

With respect to the incremental learning approaches based on SVM, the works of Cauwenberghs and Poggio (2001) and Frießet al. (1998) were two other alternatives to Syed et al. (1999). The former proposed a way to incrementally solve the global optimization problem in order to find the exact solution. Its reversible aspect allows for decremental unlearning and to efficiently compute leave-one-out estimations. The latter is the Kernel-Adatron algorithm, a very fast approach to approximate the solution of the support vector learning. It had been successfully tested to dynamically adapt the kernel parameters of the machine, doing model selection in the learning stage.

On the other hand, Zheng et al. (2012) designed an Online Incremental Support Vector Machine (OI-SVM) which mainly consists of two components: the Learning Prototypes (LP) and the Learning Support Vectors (LSV). LPs learns the prototypes and during the learning process continuously adjusts prototypes to the data concept and LSVs get a new SVM by combining learned prototypes with trained SVs. Globally, OI-SVM can effectively deal with large-scale problems, incremental problems, and online learning problems. Cauwenberghs and Poggio (2001) designed an exact on-line algorithm of incremental learning SVM, which updates the decision function parameters when adding or deleting one vector at a time. Later, Diehl and Cauwenberghs (2003) improved the previous work and presented a framework for exact SVM incremental learning, adaptation and optimization, in order to simplify the model selection by modifying the SVM solution when changing kernel parameters and doing regularization. The main drawback of most of these techniques is that they allow only binary classification. Hence, in order to tackle the problem of on-line multi-category classification, Boukharouba et al. (2009) proposed an incremental multiclass support vector classifier and the experiments showed that it could provide accurate results. Other techniques include Learning Active Support Vector Machine (LASVM) in Bordes et al. (2005), and Kivinen et al. (2004) which considered incremental learning in a Reproducing Kernel Hilbert Space using gradient descent within a feature space.

More recently, Duan et al. (2009) proposed the online and batch incremental algorithms for Lagrangian Support Vector Machine (LSVM). Their main contribution is that it is not necessary to relearn the whole database while a new sample or the new sample sets are incremented. In Liu and Yang (2010), they applied an error-driven incremental learning for the SVM classification in the traditional BoW approach. To limit the computation burden, they made use of a constraint on the number of submodels. The work of Ralaivola and d'Alche Buc (2001) exploited the characteristics of locality of RBF by re-learning only weights of training data that lie in the neighborhood of the new data. Their algorithm was designed for RBF kernel-based SVM, excluding multikernel SVM.

To go beyond the controlled environments, for large-scale problems, it is possible to scale down the problem size (down-sampling) and train with the representatives. In this context, Cluster Based Support Vector Machine (CB-SVM) proposed by Yu et al. (2003) recursively selects SV clusters along the cluster tree to get better performance; Cluster-SVM (Boley and Cao, 2004) first trains an SVM with clusters, then replaces the clusters containing only non-SVMs with its representative until each sub-cluster is SVMs or non-SVs; Finally, Super Cluster Machine (SCM) by Li et al. (2007) exploits a compatible Probability Product Kernel (PPK) which measures the similarity between clusters and between clusters and vectors.

In the end, and coming back again to the traditional SVM based approaches, Teixeira and Corte-Real (2009) made use of an ensemble based incremental algorithm with SVM as the base classifier. However, its performance was far lower than the model built with complete data. Considering that the model itself needs to store a high dimension hyperplane norm vector while the BoW vectors are very sparse with only hundreds of nonzero elements, to store part of previous data is not a very time consuming approach in comparison. On-line methods are particularly useful in the situations that involve on-line streaming data (Agarwal et al., 2008). The authors Liang and Li (2009) had proved that incremental SVM is suitable for large dynamic data and more efficient than batch SVMs in terms of the computing time. Considering these facts, a model with incremental learning SVM as a solution is implemented in our system, which, in contrast to the work of Teixeira and Corte-Real (2009), it is capable to maintain the performance obtained with the batch learning.

6.3 Improving Action Classification with an Incremental Learning

In this section we describe how information sources are incorporated within the overall system. First, we expose in Section 6.3.1 how we take advantage of BoW approach and how action objects from context information are incorporated to the

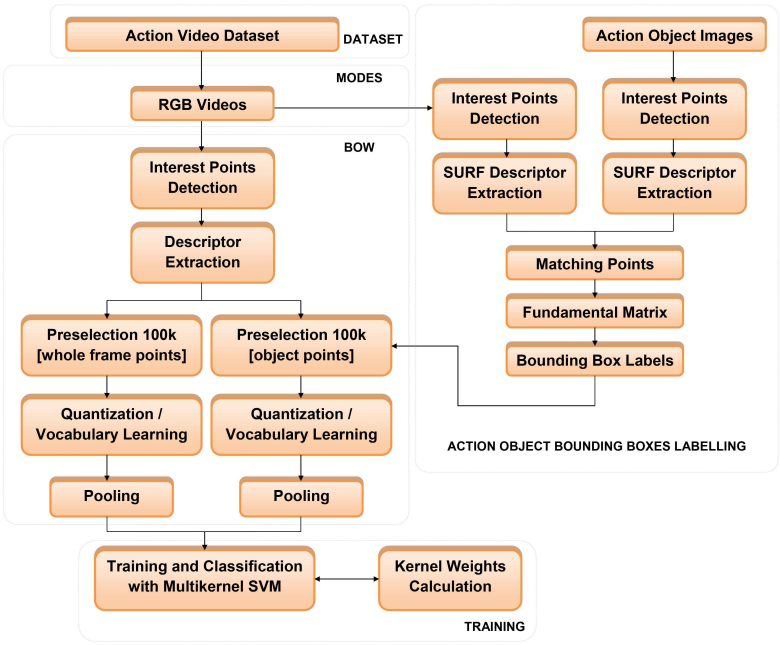


Figure 6.1: Scheme overview with action objects detection.

overall system. Second, how the different informational modes are converted to image coding by using BoW is described in Section 6.3.2. Finally, information fusion from context and modes is described in Section 6.3.3 using either a batch and incremental learning methods, exposed in detail in Section 6.3.4.

6.3.1 Feature Extraction from Contextual Information: Objects

To encode frames using BoW, STIP points following the work of Laptev (2005) are searched. Computing a descriptor for each point, a codebook is trained using a maximum of 100k randomly sampled features. Afterwards, employing the k-Means clustering algorithm the points are grouped. The size of the codebook is set following Reddy and Shah (2013) and Bilinski and Corvee (2013) where the codebook size is limited to 500 or 1000 to avoid over-learning and despite the fact that the larger the number of clusters employed, the better the performance is.

Finally, a SVM with an exponential chi-squared kernel is used for classification, combining all different BoW-based descriptors within a SVM classifier. We use a 10 fold cross-validation with one-against-all approach. An overall scheme of this

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

Figure 6.2: Object point detection and matching. Firstly, point detection and descriptor extraction for a video frame (a) and the object image (b) is released. Secondly, matches (c) and outlier filtering (d) is performed and finally, the transformed bounding box (e) is computed and labeled.

procedure can be seen in Figure 6.1

Concerning context, object information relevant to the action in RGB sequences is assembled into the BoW based representation of this action, ensuring the object point selection. To do so, we must find the object in the video sequence. Each video contains one action, and we detect action related objects. Therefore, we obtain one instance image of each object per video and use this image to find the object along the whole video by matching a set of points previously extracted from the frame and the instance image. The matching procedure, based on the epipolar geometry described in Hartley and Zisserman (2004), is described in Figure 6.2. The points are extracted using a Harris corner detector and described by SURF features. This way ensures a large set of points belonging to the object, which is necessary to obtain good point correspondences and to compute a representative bounding box. Then, we compute the point matching by applying the k-Nearest Neighbor (kNN) algorithm and setting a threshold to select the strongest matches.

Finally, we compute the fundamental matrix –excluding outliers by using Random Sample Consensus (RANSAC)(Fischler and Bolles, 1981)– and use it to obtain a transformation of the initial bounding box. This ensures more accuracy around the area that limits the object in the frame. The result of this procedure is a bounding box enclosing the object used in each action for each frame in the video, as can be seen in Figure 6.2.

In the following step, we apply the BoW based representation and select a maximum of 100k of object points employing the bounding boxes labels. In the

Table 6.1: Descriptors used to encode sensor information.

Descriptor	Characteristics	Reference
trajectories	KLT tracker or SIFT matcher	Jiang et al. (2012)
HOG	static appearance information from local gradients	Dalal and Triggs (2005)
HOF	local motion information	Lucas and Kanade (1981)
MBH	separately computes vertical and horizontal OF components	Dalal et al. (2006)
HOG3D	spatio-temporal extension of HOG	Kläser et al. (2008)
Depth_XXX	previous descriptors (trajectories, HOG, HOF, MBH) applied to depth images	- - -
FPFH	3D appearance from local gradients	Rusu (2009)

end, we construct a codebook from the pre-selected words belonging to objects and combine this codebook with others using the multikernel SVM explained in the Section 6.3.3.

6.3.2 Feature Extraction from Information Modes: RGB, Depth, 3D

RGB images are usually provided by a single camera mounted in the body of the robot or in a fixed place in the environment. This imposes the limitation of a single view of the performed action. There exist databases which consider the possibility of a multiple viewpoint, introducing more variability to the information captured. That would be the case if different robots were analyzing the same action simultaneously in different positions, but we consider human-robot interactions that involve just one robot.

As explained in the previous subsection, we used BoW frame encoding by making use of STIP points (Laptev, 2005). In this case, for RGB sequences, some descriptors can be extracted, namely, HOG3D, trajectories, HOG, HOF, and MBH, whose

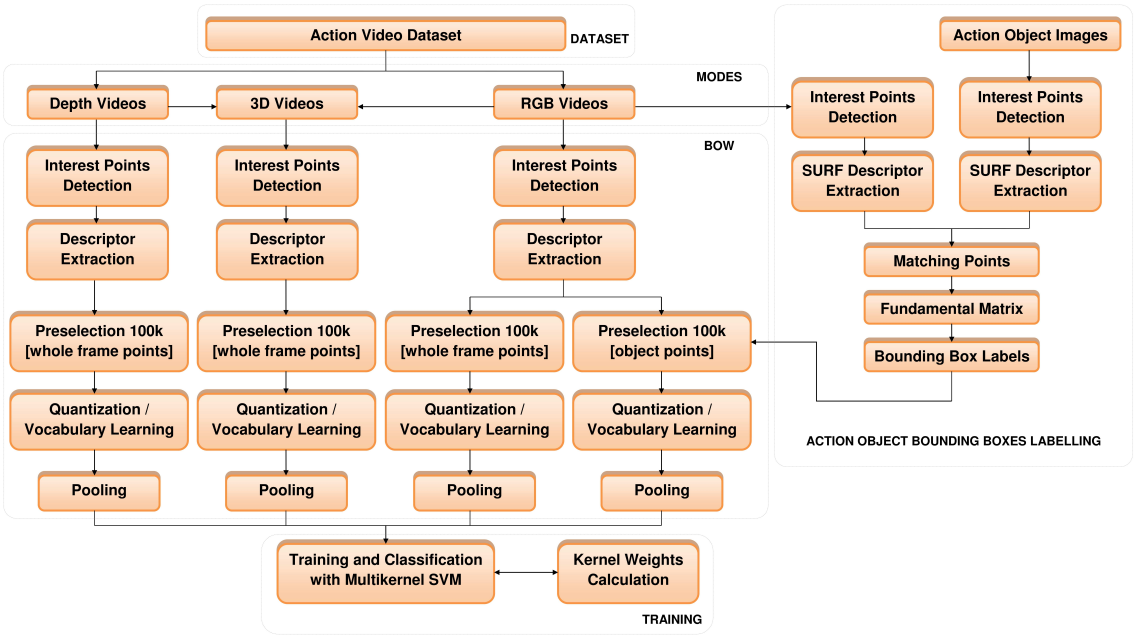


Figure 6.3: Scheme overview including contextual and modal information.

main characteristics can be seen in Table 6.1. Computing a descriptor for each point, a codebook is trained using a maximum of 100k randomly sampled features. Afterwards, employing the k-Means clustering algorithm the points are grouped. The size of the codebook is set likewise Section 6.3.1.

We make use of depth information in two ways: first, extracting descriptors as done with the RGB video sequences. We have, then, a set of descriptors such as trajectories, HOG, HOF and MBH for RGB and Depth. Depth sequences allow for the differentiation of elements in the scene like background and objects over planes different from the one in which the action takes place. Second, generating a RGB-D sequence in which we can extract 3D spatial descriptors, such as FPFH. 3D sequences provide 3D spatial information combined in one descriptor. In the end, RGB, Depth and 3D descriptors generate independent codebooks. Finally, a SVM classifier is trained for classification, combining all different BoW-based descriptors within a SVM framework. An overall scheme of this complete procedure including contextual and modal information can be seen in Figure 6.3.

In the end, our system includes RGB, Depth, 3D and action object as different channels of information. Example frames of these sources can be seen in Figure

Table 6.2: Comparison of different descriptors on the databases

Databases	**CAD120 (%)**	**HMDB (%)**
RGB trajectories	32.30	38.13
RGB HOG	70.17	54.29
RGB HOF	49.02	41.6
RGB MBH	46.97	38.30
RGB HOG3D	**83.94**	**71.98**
Depth trajectories	56.34	n/a
Depth HOG	55.99	n/a
Depth HOF	56.47	n/a
Depth MBH	55.18	n/a
FPFH	60.51	n/a

6.4. Some basic results obtained with this collection of descriptors can be seen in Table 6.2. These results show that the more complex the descriptor is, the higher the performance is. Hence, the HOG3D descriptor, which combines both space and time information has the best performance, but at the price of higher computational complexity. Furthermore, we found that spatial descriptors –HOG, FPFH– give the highest values comparing to others that give time information –trajectories, HOF, MBH– or depth description due to the amount of information inherent in those vectors.

Figure 6.4: Multimodal database CAD120 with RGB (most left), Depth map (middle left), 3D map (middle right), object context (most right).

6.3.3 Information Fusion with Weighted Contextual and Modal Multikernel SVM

Visual features extracted from a RGB video can represent a wide variety of information, such as scene (e.g., GIST (Solmaz et al., 2012)), motion (e.g., HOF (Lucas and Kanade, 1981), MBH (Dalal et al., 2006), HOG3D (Kläser et al., 2008)) or even just color (color histograms). In our approach we include extra features, such as depth and 3D scene information (e.g. FPFH (Rusu, 2009)), and object related information. To classify actions using all these features the information must be fused in an appropriate way. According to the moment of the combination, Snoek et al. (2005) proposed a classification of the fusion schemes in early or late fusion. Multikernel approaches use early fusion since the combination is done before the training.

The work of Wang et al. (2013) uses a linear combination of different kernels, calculated from a set of codebooks generated with different descriptors. A SVM with a χ^2 kernel for classification is used,

$$\chi^2(h_i, h_j) = \frac{1}{2}\sum_{k=1}^{n}\left(\frac{(h_i(k) - h_j(k))^2}{h_i(k) + h_j(k)}\right) \tag{6.1}$$

ensuring that the kernel matrices are strictly positive definite. They fuse different descriptors by summing up the corresponding kernel matrices, normalized by the average distance A^c of χ^2 distances between the training samples for the c-th channel. No kernel weighting is done, so no kernel is more discriminative than the others.

In our approach, given the base kernels

$$K_c(h_i, h_j) = exp(-\frac{1}{A^c}\chi^2(h_i^c, h_j^c)) \tag{6.2}$$

the optimal kernel of a certain descriptor is approximated as

$$K_{opt} = \sum_c d_c K_c \tag{6.3}$$

where d_c is the kernel weight for c-th channel. Each K_c represents the precoded

c-th information referred to the action.

The optimization is carried out within a SVM framework that achieves the best classification on the training set subject to a regularization scheme. In this formulation, the objective function is near identical to the standard L_1 C-SVM objective function. The regularization prevents the weights from becoming too large, although this could be achieved by requiring that the weights sum up to the unit but also restricting the search space.

$$\begin{aligned} \underset{w,d,\xi}{\text{minimize}} \quad & \frac{1}{2}w^t w + C1^t\xi + \sigma^t d \\ \text{subject to} \quad & y_i(w^t K_c + b) \geq 1 - \xi_i \\ & \xi \geq 0, d \geq 0, Ad \geq p \end{aligned} \tag{6.4}$$

The constraints are also similar to the standard SVM formulation, with the addition of two constraints. First, $d \geq 0$, which ensures that the weights can be interpreted and also leads to a much more efficient optimization problem, and second, $Ad \geq p$, with some restrictions, that allow us to encode prior knowledge about the problem.

In order to tackle large scale problems involving hundreds of kernels, we adopt the minimax optimization strategy and solve the problem by using projected gradient descent, taking care to ensure that the constraints $dn + 1 \geq 0$ and $Adn + 1 \geq p$ are satisfied. This algorithm proceeds in two stages. In the first stage, weights d_c are maximized and Support Vectors (SV) are obtained. In the second stage, objective function is minimized by projected gradient descent. The two stages are repeated until convergence or a maximum number of iterations is reached, at which point the weights d and $SV's$ are obtained.

6.3.4 Incremental Learning

Given that only a fraction of the training examples end up as support vectors (Vapnik, 2000), the SVM classifier is able to summarize the data space in a very concise manner, i.e., summarizing the data in a compact form and the selection of Support Vectors form a minimal set (Syed et al., 1999).

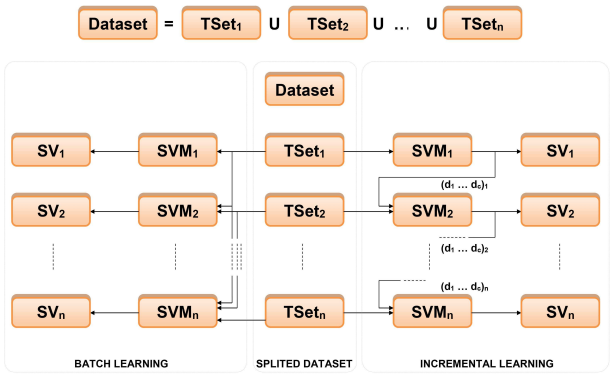


Figure 6.5: Incremental Learning workflow in comparison to the Batch Learning.

This suggests that we could partition a huge database and then incrementally train the SVM classifier with the partitions. The training would preserve only the support vectors and the classifier kernel weights at each incremental step and add them to the training set for the next step. Once discarded, the vectors from the previous training set are not considered again. The main purpose of this method is that the model obtained should be the same or similar to what would have been obtained using all the data together to train the model. The reason for this is that the SVM algorithm will preserve the essential class boundary information seen so far in a very compact form as support vectors, which would contribute appropriately to deriving the new concept in the next iteration.

In order to carry out the incremental learning the training set is divided in n subsets. We train a classifier (SVM_1) in the first training subset ($TSet_1$). From this first training, we keep the support vectors (SV_1) and the kernel weights ($(d_c)_1$), adding them to the second training subset ($TSet_2$). The weights are used to empower the discriminating capacity of the model, which gives different importance to the support vectors from each kernel. We then train a new classifier (SVM_2) and we keep the new set of support vectors (SV_2) and kernel weights ($(d_c)_2$) in order to be added to the third training subset ($TSet_3$). This incremental step is repeated until all the training subsets are used. The procedure is summarized in Figure 6.5.

6.4 Experimental Results and Discussion

In this section, a description of both databases is given in Section 6.4.1. Then, the setup of the proposed approach is explained in detail in Section 6.4.2. Finally, the results obtained are presented in Section 6.4.3.

6.4.1 Databases

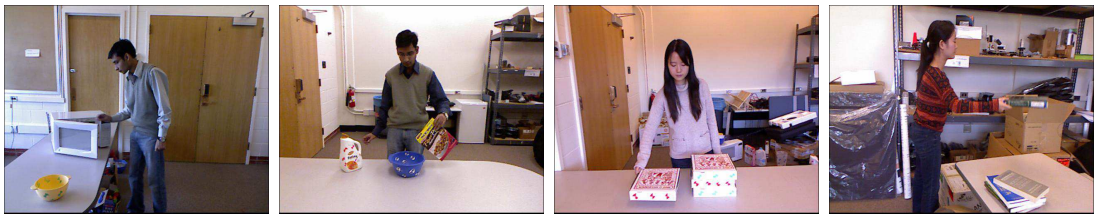
We test our model over two different databases, CAD120 (Koppula et al., 2013) and HMDB (Kuehne et al., 2011). The former contains objects that involves actions in a highly controlled environment and multimodal information such as RGB and depth videos. The latter is a more challenging and realistic one, where objects used in actions are present. Sample frames for each database are shown in Figure 6.6, in which the actions used from the whole collection are represented for both databases.

The HMDB database (Kuehne et al., 2011) consists of 51 actions from a total of 6,849 videos collected from a variety of sources ranging from digitized movies to YouTube videos. The action categories are grouped in five types: general facial actions, facial actions with object manipulation, general body movements, body movements with object interaction, and body movements for human interaction.

Considering that we need actions with object interaction, we do not follow the original splits proposed by Kuehne et al. (2011). We reduce the computational cost by pre-selecting 6 different actions with 20 videos per action, resulting in 120 videos in total. The pre-selected actions are *ride bike*, *shoot gun*, *shoot bow*, *draw sword*, *swing baseball* and *kick ball*. The purpose of this selection is dual: first, ensuring that an object is involved in the action, and second, ensuring the presence of as many variations as possible. Similar actions are also taken into account, a fact that makes the set more challenging. The split is detailed in Appendix A.

CAD120 database contains 124 RGB-D videos of 4 different subjects performing 10 high-level actions. Each action is performed three times with different objects. It contains a total of 61585 3D video frames. The actions have a long sequence of subactivities which might be considered in future work. The 10 high-level

actions performed are *arranging objects*, *cleaning objects*, *having meal*, *making cereal*, *microwaving food*, *picking objects*, *stacking objects*, *taking food*, *taking medicine* and *unstacking objects*.



(a) Microwaving food (b) Make cereal (c) Unstacking objects (d) Placing

(e) Takeout (f) Taking medicine (g) Stacking objects (h) Picking objects

(i) Eating (j) Cleaning (k) Shoot bow (l) Shoot gun

(m) Swing baseball (n) Ride bike (o) Draw sword (p) Kick ball

Figure 6.6: Example frames from both databases showing all used actions: from CAD120 database, (a) microwaving food, (b) make cereal, (c) unstacking objects, (d) placing, (e) takeout, (f) taking medicine, (g) stacking objects, (h) picking objects, (i) eating, (j) cleaning, and from HMDB database, (k) shoot bow, (l) shoot gun, (m) swing baseball, (n) ride bike, (o) draw sword, (p) kick ball.

6.4.2 Setup

The points used to identify and track the objects are a mixture of RGB points obtained using Harris corner detector and features computed applying SURF. We

use a threshold between 0,04 and 0,1 for Harris detector and a maximum number of 1000 points for SURF. This ensures enough quantity of points with enough quality belonging to the object, even in the case that the object appearing in the video sequence is relatively small, like a ball or a sword. For the matching, we select the strongest 1% of matches, which is restrictive but ensures better point correspondences. These considerations refer mainly to HMDB database, which is more realistic than CAD120. Hence, object detection and tracking for CAD120 are more accurate due to their highly controlled conditions.

We select three informational modes taking advantage of the RGB-D videos, forming the set with RGB, depth and 3D videos.

First, for each point in RGB and Depth videos we compute different descriptors, HOG3D, trajectories, HOG, HOF, MBH. In the case of HOG3D descriptors, we set the parameters optimized for KTH database as described in Kläser et al. (2008), which have demonstrated a good performance not only for the KTH set, resulting in 1008 dimensions in total. In the case of trajectories, HOG, HOF, and MBH, we follow the work of Wang et al. (2013) and set the parameters likewise. The dimensions of these descriptors are, respectively, 30 (trajectories), 96 (HOG), 108 (HOF) and 192 (MBH), which are significantly smaller than those of HOG3D. We set the same parameter values for both RGB and Depth videos.

Second, we consider the FPFH descriptor (Rusu, 2009) of the 3D Point Cloud Library. We configure the descriptor length to FPFHSignature33, that creates a 33 dimension descriptor. We set the FPFH radius search to 100 in order to ensure enough valid descriptors.

We use the BoW approach to encode frames. First, we make use of STIP points following the work in Laptev (2005). We compute different descriptors for each point in RGB videos, Depth videos and 3D videos. We train a codebook for each descriptor type using a maximum of 100k randomly sampled features. For the object kernel, we ensure the object point selection using the method described in Section 6.3.1.

Afterwards, we group the points employing the k-Means clustering algorithm with a maximum of 5 iterations which ensures enough convergence. According to Reddy

and Shah (2013), the size of the codebook is set to 500 words, avoiding over-learning, despite the fact that the larger the number of clusters employed, the better the performance is. Finally, a multikernel SVM as described in Section 6.3.3 is used for classification, using a 10 fold cross-validation method with the one-against-all approach. For all the experiments we employ the default parameter values in the LibSVM library (Chang and Lin, 2011).

To evaluate the effectiveness of incremental training, we compare the performance of the incrementally trained model against the model trained with all the data so far in one batch. More specifically, we need to compare the following two cases.

The first case is where we have only the learned model from the previously seen data (preserved in the form of support vectors and kernel weights), and as new information comes in, we want to know how does the classification algorithm do, i.e., if the performance deteriorates as we keep discarding the redundant examples at each successive incremental step, when the new examples come in.

In the second case, we save all the previously seen data, and as new information arrives, we want to know how well does the classification algorithm (SVM) fare on unknown data. This is effectively batch learning using all the data seen so far.

We perform experiments with 0, 2, 5, and 10 splits of the training set, where 0 means a batch learning. We compare the results obtained with all these four experiments.

6.4.3 Recognition Outcome

The use of Contextual and Modal MultiKernel Learning Support Vector Machine (CMMKL-SVM) developed in Chapter 5 allows the addition of different descriptors into the standard BoW approach for action recognition. This approach permits the inclusion of several image descriptors into this scheme as explained in Section 6.3.3, and reduces the effect of information redundancy weighting a multikernel SVM. This approach improves the performance with respect to any singular descriptor or an averaged combination of them.

In our first experiment we calculate the average accuracy for each of the following

descriptors: trajectories, HOG, HOF, MBH, Depth_trajectories, Depth_HOG, Depth_HOF, Depth_MBH, Depth_HOG3D and FPFH. As we can see in Table 6.2 from Section 6.3.2, HOG3D descriptor gives the best action recognition performance. HOG3D avoids non-trivial pre-processing steps, such as tracking and segmentation, fuses 2D space and time information, and provides descriptors invariant to illumination and camera motion. This aspect shows that using a unique optimal descriptor can be better than a combination of several descriptors that perform worse individually. This is apparent in the fact that HOG3D obtains a 71.98% for HMDB and 83.94% for CAD120.

An extra objective of our approach is to surpass this performance obtained with HOG3D descriptor by using CMMKL-SVM with the best weighted combination of descriptors using RGB videos, Depth videos, 3D points and objects. That would considerably reduce the time of the overall procedure, taking into account that HOG3D is quite computationally expensive. Additionally, Depth descriptors give similar results (55%), for each single descriptor -trajectories, HOG, HOF and MBH- meaning that these descriptors loose their singular characteristics when used for depth videos. On the other hand, HOG and FPFH are the best choices when used as single descriptors, obtaining a recognition rate of 70.17% and 60.51% respectively in CAD120. This is due to the fact that they give the spatial information of the action, a fact that has been verified in Chapter 4.

In the second experiment, our purpose is to observe the influence of the context (objects) and mode (Depth, 3D) when employing single descriptors (trajectories, HOG, HOF, MBH) on RGB videos. The results are shown in Table 6.3. We perform the experiments with our approach CMMKL-SVM and we show that the fusion of context and mode information in a MKL framework improve the recognition performance. Having a look at the results we can observe that the addition of context, gives an important improvement of 20% on the average recognition rate for every trial in HMDB when using our approach. For CAD120, this improvement is significantly lower than for HMDB, 10% on average, due to the quality of the videos and the lack of extensive variability in conditions such as illumination and viewpoint.

In Table 6.3 we show how context, depth or 3D information always outperforms the recognition accuracy reached using a single RGB based descriptor.

Table 6.3: Context and modal influence on the databases using batch learning (CMMKL)

CAD120 Database	**CMMKL (%)**	**Kernel Weights**
Object information with RGB descriptor		
obj + RGB trajectories	56.57	0.5/0.7
obj + RGB HOG	86.32	0.2/0.8
obj + RGB HOF	74.94	0.4/0.8
obj + RGB MBH	68.32	0.4/0.6
Depth information with RGB descriptor		
Depth + RGB trajectories	83.19	0.7/0.3
Depth + RGB HOG	89.59	0.9/0.9
Depth + RGB HOF	86.53	0.4/0.8
Depth + RGB MBH	87.96	0.1/0.4
3D spatial information with RGB descriptor		
FPFH + RGB trajectories	87.98	0.6/0.4
FPFH + RGB HOG	90.67	0.2/0.5
FPFH + RGB HOF	89.28	0.7/0.7
FPFH + RGB MBH	90.18	0.8/0.7
HMDB Databases	CMMKL (%)	Kernel Weights
Object information with RGB descriptor		
obj + RGB trajectories	55.43	0.1/0.2
obj + RGB HOG	86.72	0.5/0.7
obj + RGB HOF	65.37	0.3/0.4
obj + RGB MBH	61.61	0.8/0.5

Furthermore, in Table 6.4, we can see the importance of weighting channels, which takes into account the redundancy of information introduced by similar descriptors. HOG remains the most significant descriptor and this reinforces the hypothesis that the strongest descriptors are those that provide spatial information.

Finally, considering the fusion of space, time, depth, 3D and objects, we perform an experiment which could give us feedback about incremental learning. We took trajectories, HOG, Depth_HOG, FPFH and object descriptors in order to obtain the performance using a batch learning and using an incremental learning, which was characterized by using three different numbers of subsets: $n = 2$, $n = 5$ and $n = 10$.

The results in Table 6.4 also show the evolution of the performance in each case. When starting, the performance is high because the amount of data is limited and as this amount gets larger, the performance become increasingly smaller.

Table 6.4: Performance and Kernel weights evolution through the incremental learning when $n = 0$, $n = 2$, $n = 5$ and $n = 10$, where $n = 0$ equals to batch learning. In order to get the best performance we combined all information sources available in each database, i.e., trajectories, HOG, FPFH, Depth_HOG, and objects for CAD120 and trajectories, HOG and objects for HMDB.

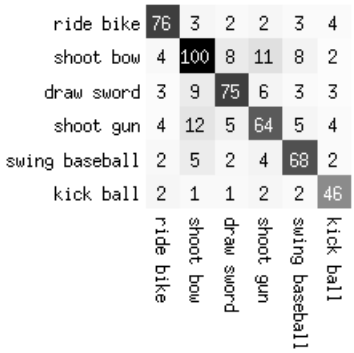
Databases	**CAD120**		**HMDB**	
# subsets	**IWCMMKL (%)**	**Kernel Weights**	**IWCMMKL (%)**	**Kernel Weights**
$n = 0$	**92.83**	**0.6/1.0/0.8/0.1/0.5**	**85.84**	**0.3/0.7/0.2**
$n = 2$	96.47	0.6/0.9/0.1/0.2/0.5	92.73	0.5/0.4/0.4
	89.39	**0,8/0,9/0,8/0,5/0,8**	**87.12**	**0.2/0.6/0.1**
$n = 5$	95.44	0.7/0.6/0.5/0.1/0.2	96.93	0.3/0.3/0.3
	97.84	0.3/0.4/0.7/0.4/0.5	96.38	0.8/0.4/0.1
	96,29	0.1/0.1/0.1/0.4/0.9	94.62	0.9/0.1/0.4
	95.82	0.8/0.5/0.8/0.1/0.7	92.73	0.5/0.4/0.1
	89.12	**0,4/0,9/0,8/0,1/0,9**	**87.86**	**0.1/0.9/0.1**
$n = 10$	88.56	0.2/0.4/0.5/0.1/0.6	96,66	0.1/0.6/0.2
	95.80	0.4/0.5/0.6/0.3/0.7	96,78	0.6/0.7/0.7
	98.49	0.7/0.2/0.7/0.4/0.4	96,33	0.1/0.5/0.9
	96.81	0.5/0.8/0.6/0.5/0.3	94,84	0.2/0.1/0.2
	97.39	0.9/0.1/0.1/0.4/0.8	94,96	0.3/0.8/0.1
	98.21	0.8/0.2/0.2/0.6/0.9	87,74	0.1/0.1/0.9
	97.82	0.2/0.6/0.5/0.3/0.9	92.93	0.9/0.1/0.5
	97.24	0.8/0.2/0.7/0.3/0.3	91.61	0.8/0.3/0.7
	95.49	0.4/0.3/0.2/0.9/0.3	90.27	0.1/0.9/0.1
	92.54	**0,6/0,8/0,6/0,2/0,6**	**87.50**	**0,80/0,17/0,94**

What is relevant in our approach is that thanks to the recursive adaptation of the kernel weights, the final percentage obtained is always equivalent to the one obtained with batch learning. Thus, considering CAD120 database results, batch learning value is 92.83%, for $n = 2$ the value is 89.39%, for $n = 5$ is 89.12% and for $n = 10$ is 92.54%. On the other hand, considering the HMDB database, those vales are 85.84%, 87.12%, 87.86% and 87.50% respectively. The results are summarized in Table 6.5 and they show that with our approach the learning does not degrade the performance obtained by all the trained data as a whole, therefore allowing the incorporation of new data without having to learn from batch again.

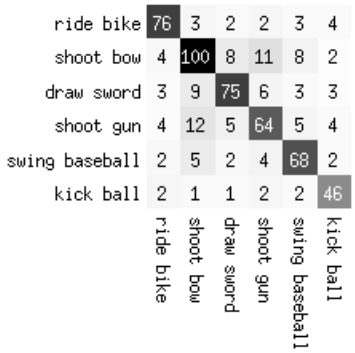
Table 6.5: Comparison between different subset number n on training set. We took $n = 0$, $n = 2$, $n = 5$ and $n = 10$ for our experiments, where $n = 0$ equals to batch learning. Experiments were repeated 10 times in order to see the consistency and repeatability of the results.

Database	CAD120	HMDB
$n = 0$	92.83 ± 1.23	85.84 ± 1.19
$n = 2$	89.39 ± 1.61	87.12 ± 0.96
$n = 5$	89.12 ± 1.28	87.86 ± 1.05
$n = 10$	92.54 ± 1.42	87.50 ± 1.30

To summarize, in terms of the confusion of the actions, Incremental Weighted Contextual and Modal MultiKerneL Support Vector Machine (IWCMMKL-SVM) reduces confusion between actions, even for similar actions, as can be seen in Figure 6.7 and 6.8. For example, *Unstacking objects* for CAD120 is easily confused with *Stacking objects*, a relation that the approach breaks (32%). In general, all actions in both databases have their confusion index reduced. What is important to see in the Figure 6.7 and 6.8 is that when performing the experiments with the incremental approach IWCMMKL-SVM the relation between similar actions remain broken, and the confusion rate is similar to the one obtained using a batch approach.

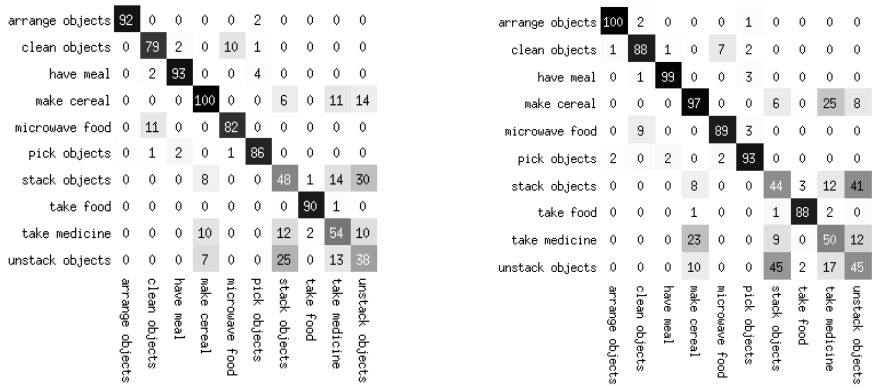


(a) HMDB with Batch learning



(b) HMDB with Incremental learning (n=5)

Figure 6.7: Confusion matrices for HMDB database using objects, trajectories and HOG, descriptors, with an average performance of (a) 85.84% on batch learning and (b) 87.86% using incremental learning with 5 subsets.



(a) CAD120 with Batch learning

(b) CAD120 with Incremental learning (n=5)

Figure 6.8: Confusion matrices for CAD120 database using objects, trajectories, HOG, Depth_HOG and FPFH descriptors, with an average performance of (a) 92.83% on batch learning and (b) 89.39% using incremental learning with 5 subsets.

6.5 Summary

We have introduced a method to incrementally add new information to the trained system and we have demonstrated that it has no significant loss in the performance when compared to the batch training. The system splits the training set into equally divided subsets and successively trains each subset with the support vectors from the classifier trained with the previous one. We take advantage of the fact that Support Vector Machines are able to summarize the data space in a compact form and the selected Support Vectors form a minimal set.

The experiments were carried over two relevant databases, CAD120 and HMDB. The former includes Modal information, from depth sensors particularly, and both databases include actions with object interaction. This fact allows us to perform the experiments using a fusion of information sources, i.e., space-time from RGB videos, depth, 3D space from a previously produced 3D sequences and objects used to perform the action by computing its bounding boxes. The results show that when using a batch learning the weight of the different kernels allows the proper discrimination of the redundant information and that when the incremental approach is used, those weights change its value because they have been calculated recursively and the redundant information has been filtered in n steps. This fact

avoids the deterioration of the performance obtained with our approach, allowing the incorporation of new data without having to compute a training with the whole amount of data.

Considering that the approach allows the addition of new data without a loss of the overall performance, the next steps would be to improve this approach by allowing the incorporation of new class data. This means that new actions could be demonstrated to a robot and it would be able to learn incrementally in data and actions.

Chapter 7

Discussion and Further Work

"All we have to decide is what to do with the time that's given to us."

- J.R.R. Tolkien, *The Fellowship of the Ring*

7.1 Outline of the chapter

In this thesis, a general method to recognize actions that need to be imitated is developed. It responds to the question of *What to imitate?* present in PbD. The model is trained over large datasets and validated over subsets of them, and it is capable to differentiate similar actions considering the multiple information sources that a robot have, such as depth sensors, visual sensors, and context. Furthermore, the recognition engine can recognize both objects and actions present in a video sequence and can intensively be used in a variety of robot applications, such as VinBot and RoboHow projects, which are explained in detail in Appendix C.

In addition, this dissertation introduces a method to incrementally add new information to the trained system based on the fact that Support Vector Machines are able to summarize the data space in a compact form and that the selected Support Vectors form a minimal set. The model ensures high accuracy, reversibility and high rate of action differentiation.

This final chapter presents a summary of the contributions and final remarks of this thesis and suggests future research directions.

7.2 Summary of Contributions

As seen in Chapter 2, the selection of one PbD approach or another must take into account various steps. First of all, we must consider what information provided by the instructor during a demonstration will be recorded and used to teach our robot learner. Choosing one particular approach will be determined, for example, by whether the capturing system can be used directly on the instructor or whether the information can be obtained from the cameras looking at the scene. There are several possibilities and the actual task to be learned will determine to a great extent the most suitable way of capturing the movements.

Creating a set of policies to control the robot depends on the type of information we obtain from the sensors and the quality of this information, and also the algorithm that performs the learning process. Data can be continuous or discrete, but it also has different levels of quality, from motor commands to higher semantic behavioral levels, such as *'bring an apple from the kitchen'*. On the other hand, considerations about the construction of the learning algorithms –e.g., the mapping functions– and at what precise moment the resulting actions will be required also determine the approach to chose.

A series of drawbacks must be taken into account when designing the PbD procedure. Since the learning process usually consists of iterative algorithms which will adjust the results according to some objective function until a goal is met, it is important to keep several measures of performance in mind in order to reach the appropriate solution, as for example, the total time and convergence speed of a particular algorithm, the robustness to perturbations and precision of the solutions obtained in the demonstrations and the possibility of adding new examples to the learning process. In this respect, stability is a key element for most current approaches. However, stability can sometimes only be maintained locally, while at other times it can be maintained globally. Two of the most important aspects that generate low performance are under-demonstrated states, which turn into poor generalizations and problems when new data is incorporated, and when data is of

poor quality from ambiguous and irrelevant demonstrations.

Looking at the taxonomy of the leading current approaches to PbD discussed in Chapter 2, we can appreciate that the two important steps that need to be resolved are the measure of the performance of the result and the procedure to model the dynamic of the system with respect to the learning algorithm. For the former, most approaches use norms that measures the discrepancy with respect to the ideal demonstration provided by the instructor. The advantage of this approach is that we can rely on the instructor's experience and goodwill, but the dependence on the instructor might also limit the overall performance of the system if the instructor is less competent or tasks are more complex and it is more difficult to give correct demonstrations. For the second problem, the main limitation is the scope of the task, i.e., a combination of primitives that can be learned independently and generalized or a longer task involving more extensive behaviors. This might compromise not only the stability of the solution but also the codification of the task into primitive ones, if there are large differences between instructor and learner embodiment.

7.2.1 A New Methodology for Action Recognition

In Chapter 3 we analyzed and compared three different methods for computing the codebook of the traditional BoW approach. We used the final action recognition performance as the evaluation metric and, finally, we evaluated our framework using a public action database. We discussed the importance of selecting the clustering parameters and determining their influence on the recognition results. In the end, we considered using the random selection, with which recognition performance was surprisingly good.

In Chapter 4 we proposed a method to incorporate action contextual information that extends a previous method used to combine motion related information into a standard action recognition scheme based on BoW. This approach allows the addition of information related to the tool or object employed in the execution of an action and shows an increment of the overall recognition performance. We have shown that adding information without any specific purpose might lead to a lack of improvement

adding the consequent computational cost to the scheme. Our approach complements space and time information and proposes a procedure to add any sort of contextual information that can be further generalized to include other data apart from the object used during an action. Additionally, the approach shows that the best results are obtained when kernels from spatial, temporal, and tool information are combined into a multichannel SVM kernel.

In Chapter 5 we proposed a methodology to combine different descriptors within a standard action recognition scheme based on BoW. The approach adds action contextual information about objects, depth maps and 3D points, and shows an increment in overall action recognition performance. The addition of the extra image descriptors, either from RGB context or sensor modality, increases the computational cost, so it is important to discriminate, or even discard, the less important descriptors. Our approach complements space and time information extracted with video descriptors, and proposes a procedure for incorporating and weighting any contextual and modal information that can be further generalized to include other data provided by new context descriptors and/or new devices. The approach also shows that results are best when kernels from spatial, temporal, context, 3D points and depth are combined within the CMMKL-SVM approach. In this respect, recognition rates were highest when a combination of trajectories, HOG, FPFH, Depth_HOG and object is used.

In Chapter 6 we have introduced a method to add new information incrementally to the trained system and we have demonstrated that it has no significant performance loss when compared to the batch training. The system splits the training set into equally divided subsets and successively trains each of them with the support vectors from the classifier trained with the previous one. We take advantage of the fact that Support Vector Machine can summarize the data space in a compact form and the selected Support Vectors form a minimal set. The results show that when a batch learning is used the weights of the different kernels make it possible to discriminate the redundant information and that when the incremental approach is used, these weights change their value because they have been calculated recursively

and the redundant information has been filtered in n steps. This means that the performance obtained with our approach does not deteriorate, and makes it possible to incorporate new data without having to compute a training with all the data.

7.2.2 Application to Public Databases

The most preliminary experimentation was carried out on the KTH database in which basic variations are employed, i.e., scene, person, illumination and camera distance. This database was used to check that the results we were obtaining made sense and to compare our approach with a commonly used database. However, we could not experiment with the KTH database for too long because of its simplicity. Hence, we also experimented on two relevant databases, CAD120 and HMDB. The former contains modal information, particularly from depth sensors, and both databases contain actions with object interaction. This allows us to merge information sources, i.e., space-time from RGB videos, depth, 3D space from previously produced 3D sequences and objects used to perform the action by computing their bounding boxes. The latter is a highly challenging database because of its realistic videos.

7.2.3 Real Applications to Innovation Projects

Video demonstrations of robot-assisted procedures can be used for Learning from Demonstration (LfD), developing finite state machines, assessing surgical skills, and calibrating. Due to the industrial nature of this dissertation, we used our approach in two real situations: VinBot and RoboHow. Both of these projects needed to tackle a learning process and they are presented in the Appendix C.

The VinBot project had two problems in facing navigation and grape production. They are both recognition problems because robots have to avoid obstacles when navigating and to recognize grapes along the canopy row in order to calculate production. Our approach propose an object recognition framework based on different information sources that include context and modes.

In the RoboHow project, we provide visual information to a kinesthetic teaching framework in order to auto-segment the high level task into atomic actions.

Robot kinesthetic and dynamic information were not enough to perform this auto-segmentation task, so we used the action recognition engine of our approach.

7.3 Future Lines of Research

The concepts and the results presented in this dissertation pave the way for new applications and solutions to different action learning and imitation learning problems. Some future research directions are summarized below.

7.3.1 Programming by Demonstration

With respect to the future of Programming by Demonstration (PbD), Cangelosi et al. (2010) proposed a roadmap of action learning research starting in 2010 and continuing for 20 years. Until 2012, Programming by Demonstration (PbD) focused on how to solve action learning, using only the simplest actions or movements, intended as complete motor primitives. At present, we are working on the second milestone in the roadmap: the flexible acquisition of action patterns and their combination to achieve more complex goals. For further details one can refer to the work by Karaoguz et al. (2013) or to Mülling et al. (2013), who put forward the idea that complex motor tasks could be tackled using several movement primitives.

The acquisition of hierarchical and compositional actions is expected to be solved in the coming years, and by 2016, the association between syntactic constructions and composite actions via social learning is likely to be the main focus of the investigation. Future developments of Programming by Demonstration (PbD) might also consider the semantic content of human commands, which can be found in natural language, but specially in visual content provided by cameras or instructional videos.

7.3.2 Task Sharing

Since this dissertation faces the first question in PbD, *what to imitate?*, the actions can be shared through the cloud. Furthermore, the robots will be able to download knowledge about new actions that have been previously learned by other robots.

With this purpose, an action recognition algorithm could be built on the cloud allowing users to update its knowledge with the addition of actions learned locally. This fact would build a new paradigm to the embodiment problem and, more specifically, to *how to imitate?* question in PbD.

7.3.3 Action Recognition

Due to the relevant importance to intelligent robots, our future work will focus on improving multimodal fusion and reducing the computational burden by using different optimization techniques for Multiple Kernel Learning (MKL), so that robots can respond more quickly in their interaction with humans by either imitating or anticipating actions.

Considering complex tasks which encloses a sequence of atomic actions, to improve the accuracy of the action recognition algorithm and make it more robust, it is possible to take into account the transitions between the atomic actions, since they are recorded in a sequential order.

7.3.4 Incremental Learning

Because the approach presented in this dissertation enables new data to be added without any loss in the overall performance, the next steps would be to improve the approach by allowing new class data to be incorporated, which means that new actions could be demonstrated to the robot and it would be able to learn incrementally in data and actions.

Learning incrementally allows the robot to learn from its own performances in addition of acquiring new demonstrations from the action performer. As a benefit of this fact, the robot would improve the task performance by acquiring new demonstrations from other performers than the original, considering that the parameters such as illumination, viewpoint, occlusions, etc. would be changed extensively. As a second benefit, the robot would be able to be taught to perform new actions until a limit of class number is reached.

Appendix A

Public Databases

When action recognition started to become a topic of interest, single camera databases were used to classify actions with a human performer. Parameters like color, texture, viewpoint, zoom, focus, environment and performers were first considered in KTH database. Furthermore, more challenging databases were created in order to introduce parameters such as human body occlusions, camera motion, video quality, and number of actions performed. HMDB and UCF databases are two of the most challenging databases today for action recognition topic.

Table A.1: The most relevant databases from the beginning till today. Basic features.

DATABASE	YEAR	♯ VIDEOS	♯ ACTIONS/ SUBACTIONS	♯ ACTORS	SCENES
KTH	2004	2391	6	25	in/out
UCF Sports	2008	150	9	-	in/out
HMDB51	2011	6849	51	-	in/out
TUM Kitchen	2009	17	4	4	in
CAD120	2011	120	10/10	4	in
YouCook	2013	88	6/6	-	in
MHAD	2013	660	11	12	in
KIT	2015	>3704	15	49	in
CMU-MMAC	2008	2605	5	43	in

The inclusion of other sensor information in the database allowed the authors to focus efforts in how to combine all those sources and get not only the better performance in action recognition but also the goal of action inference and imitation. To this purpose, Cornell University created CAD60 and CAD120 action databases,

in which RGB-D cameras, MoCap and skeleton information are present. More recent databases incorporate Audio (Natural Language (NL) descriptors), Radio Frequency Identification (RFID) tag reader, Master Motion Map (MMM), Inertial Motion Unit (IMU), accelerometer/gyroscope/magnetometer, wearable. None of them still incorporates kinesthetic and dynamic information from the robot and most of them are used to recognize actions involving one person or robot. A comparison between the most interesting and relevant databases is attached in Tables A.1 and A.2.

Table A.2: The most relevant databases from the beginning till today. Advanced features.

DATABASE	MODALITIES	♯ VIEWS	CAMERA MOTION	GROUND TRUTH	INTER-ACTION
KTH	B/W	1	static	action	-
UCF Sports	RGB	1	several	action	-
HMDB51	RGB	1	several	action/♯act/ body parts/ cam.motion/ video quality	H-H, H-obj
TUM Kitchen	RGB, MoCap, RFID tag read., magneto	12	static	action/ body parts	-
CAD120	RGBD, MoCap, Skeleton	1	-	H.lv.action/ subaction/ obj.afford.	H-obj
YouCook	RGB, NL descriptors	1	several	Action/obj./ NL desc.	H-obj
MHAD	RGBD, MoCap, Audio, acc.	1	static	action	-
KIT	RGB, MMM, MoCap	4	static	action/ anth. parts	H-obj
CMU-MMAC	RGB, MoCap, Audio, IMU, wearable	6	static	action	-

We test our model over three different databases, KTH (Schuldt et al., 2004), CAD120 (Koppula et al., 2013) and HMDB (Kuehne et al., 2011). The first database does not contain any tool or object related to any action. Despite we can not take advantage of any contextual data, this experimentation allows us to test whether our approach is comparable to these of the state of the art. The second contains objects that involves actions in a highly controlled environment and multimodal information such as RGB and depth videos. The last is a more challenging and realistic one,

where objects used in actions are present.

A.1 KTH Database

The KTH database (Schuldt et al., 2004) consists of 6 actions performed by 25 actors in a structured homogeneous environment with a total of 600 videos. As can be seen in Figure A.1, the actions performed are *boxing*, *hand-waving*, *hand-clapping*, *running*, *walking* and *jogging*, with no object involved in any of these actions. In order to reduce the computational burden, we pre-select 12 videos for any action performed by randomly selected actors into different environments, ensuring that as many variation as possible are employed, i.e., scene, person, illumination and camera distance, which makes a total of 72 videos. The four different scenarios are: outdoors *s1*, outdoors with scale variation *s2*, outdoors with different clothes *s3* and indoors *s4* as illustrated in Figure A.2.

(a) Boxing (b) Hand waving (c) Hand clapping

(d) Running (e) Walking (f) Jogging

Figure A.1: Example frames from KTH database, (a) boxing, (b) hand waving, (c) hand clapping, (d) running, (e) walking, (f) jogging.

Each file contains about four subsequences used as a sequence in our experiments. The subdivision of each file into sequences in terms start_frame and end_frame as well as the list of all sequences is given in an annotation text file.

(a) p05_s04 (b) p14_s01 (c) p22_s02 (d) p24_s03

Figure A.2: Variations in scene (s01-s04), person (p01-p25), illumination and distance for the hand clapping action.

A.2 HMDB Database

The HMDB database (Kuehne et al., 2011) consists of 51 actions from a total of 6,849 videos collected from a variety of sources ranging from digitized movies to YouTube videos. The action categories are grouped in five types: general facial actions, facial actions with object manipulation, general body movements, body movements with object interaction, and body movements for human interaction.

(a) Shoot bow (b) Shoot gun (c) Swing baseball

(d) Ride bike (e) Draw sword (f) Kick ball

Figure A.3: Example frames from actions chosen in HMDB database, (a) shoot bow, (b) shoot gun, (c) swing baseball, (d) ride bike, (e) draw sword, (f) kick ball.

Considering that we need actions with object interaction, we do not follow the original splits proposed by Kuehne et al. (2011). Additionally, we reduce the computational cost by pre-selecting 6 different actions with 20 videos per action, resulting in 120 videos in total. As illustrated in Figure A.3, the pre-selected actions

are *ride bike*, *shoot gun*, *shoot bow*, *draw sword*, *swing baseball* and *kick ball*. The purpose of this selection is dual: first, ensuring that an object is involved in the action, and second, ensuring the presence of as many variations as possible. Similar actions are also taken into account, a fact that makes the set more challenging.

In order to ensure the presence of as many variations as possible, we follow a proportion of clips similar to that in the complete database. The whole set of videos corresponding to these 6 actions has a 63.44% of actions showing the *full body*, a 32.51% showing the *upper body*, a 2.46% the *head*, and a 1.59% the *lower body*. The set we selected has a proportion of 63.33%, 32.5%, 2.5% and 1.67% respectively. We also maintain the same proportions for the number of people involved (*1*,*2*, *other*), camera motion (*motion*, *no motion*), camera viewpoint relative to the author (*front*, *back*, *left*, *right*) and for the video quality (*bad*, *medium*, *good*). All the values of these proportions can be seen in Table A.3.

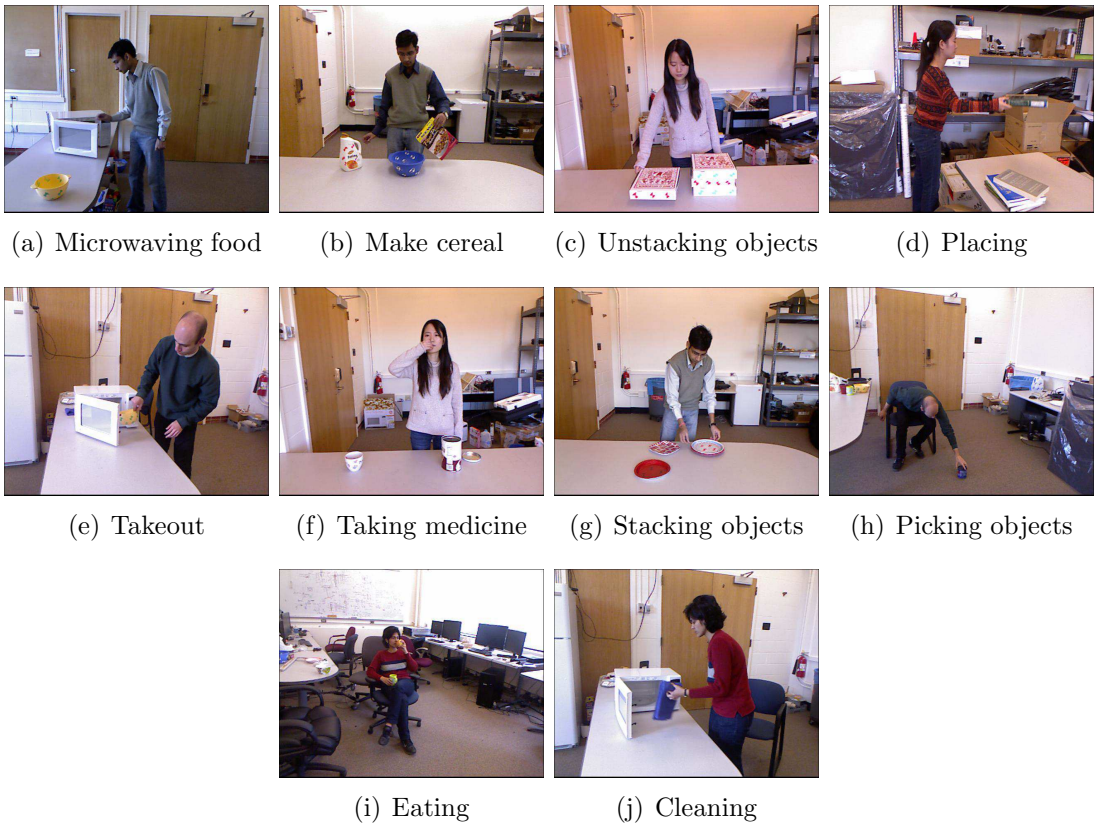
Table A.3: HMDB subset selection. We maintain proportions with respect to the original set of videos for the same actions: ride bike, shoot gun, shoot bow, draw sword, swing baseball, and kick ball.

		Original set (%)	our own set (%)
part of body	face	63.44	63.33
	head	1.59	2.5
	legs	2.46	1.67
	upper	32.51	32.5
♯ people	np1	92.77	90.83
	np2	4.19	3.33
	npn	3.04	5.84
camera motion	cm	52.46	60.83
	nm	47.54	39.17
camera viewpoint	back	18.06	20
	front	49.28	46.66
	left	16.91	16.67
	right	15.75	16.67
video quality	bad	19.80	21.67
	good	8.24	9.16
	med	71.96	69.17
♯ videos		692	120

A.3 CAD120 Database

CAD120 database (Koppula et al., 2013) contains 124 RGB-D videos of 4 different subjects –two male, two female, one left-handed – performing 10 high-level actions. Each action is performed three times with different objects. It contains a total of 61585 3D video frames. The actions have a long sequence of subactivities which can be considered as a new line of research for future work. Although it is not used in this thesis, affordability labels and tracked skeletons are also present in the database.

The 10 high-level actions performed are *arranging objects*, *cleaning objects*, *having meal*, *making cereal*, *microwaving food*, *picking objects*, *stacking objects*, *taking food*, *taking medicine* and *unstaking objects*. A single frame for each action can be seen in Figure A.4.



(a) Microwaving food (b) Make cereal (c) Unstacking objects (d) Placing

(e) Takeout (f) Taking medicine (g) Stacking objects (h) Picking objects

(i) Eating (j) Cleaning

Figure A.4: Example frames from CAD120 database, (a) microwaving food, (b) make cereal, (c) unstacking objects, (d) placing, (e) takeout, (f) taking medicine, (g) stacking objects, (h) picking objects, (i) eating, (j) cleaning.

We make use of depth information in two ways: first, extracting descriptors from

depth sequences which allow to differentiate elements in the scene like background and objects over planes different from the one in which the action takes place. Second, generating a RGB-D sequence in which we can extract 3D spatial descriptors, such as FPFH. 3D sequences provide 3D spatial information combined in one single descriptor. Frames from RGB, Depth and 3D sequences are shown in Figure A.5 as well as the object detection.

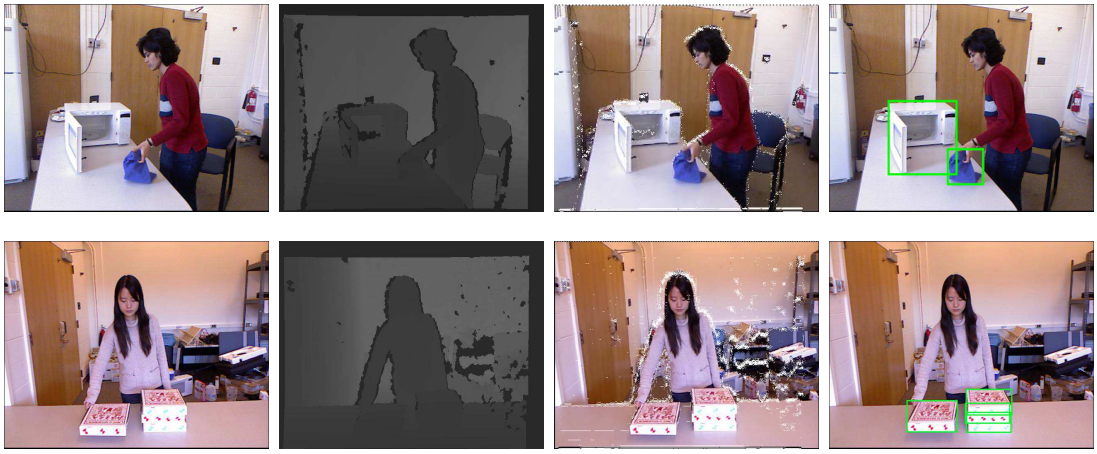


Figure A.5: Multimodal database CAD120 with RGB (most left), Depth map (middle left), 3D map (middle right), object context (most right). Modes for two actions are shown: cleaning and unstacking.

Appendix B

Publications of the Author

This thesis is supported by the publications of the author listed in this appendix with a brief comment on how they are connected to this dissertation. The publications are sorted by the first submission date.

Bautista-Ballester, J., Vergés-Llahí, J., Puig, D.: Programming by Demonstration: A Taxonomy of Current Relevant Methods to Teach and Describe New Skills to Robots, *ROBOT2013: First Iberian Robotics Conference*, Advances in Robotics, Vol. 1, Part V, pp 287-300, Springer International Publishing. 2014.

With this publication, we intended to understand how to employ the PbD paradigm for the tasks of skill learning and transference in the context of networked autonomous mobile robots. As it is shown in the paper, PbD is a natural approach to deal with both the problems of learning skills from demonstrators and the representation of skills among different robotic embodiments. Despite most of the approaches analyzed in the paper were usually applied to more human-like platforms, such as humanoids or robotic arms, we also wanted to investigate what type of approaches best fitted our specific mobile robot platform from Vinbot project.

Bautista-Ballester, J., Vergés-Llahí, J., Puig, D.: Clustering Analysis for Codebook Generation in Action Recognition using BoW Approach, In

***Proceedings of the XV Workshop of Physical Agents*, León, Spain, June 2014.**

The point of this work was to show the influence of different methods for clustering information extracted from the image, i.e. to build a good codebook when using the traditional BoW approach. Furthermore, to find why K-means algorithm is widely used for codebook generation, instead, for example, random selection of the centres.

Bautista-Ballester, J., Vergés-Llahí, J., Puig, D.: Using Action Objects Contextual Information for a Multichannel SVM in an Action Recognition Approach based on Bag of Visual Words, In *Proceedings of the 10th International conference in Computer Vision Theory and Applications*, Berlín, Germany, March 2015.

The contributions of this paper were the introduction of contextual information of actions into BoW-based description of video frames and the recognition structure that allows the addition of new information source using multikernel classifier. The basis of this paper is explained in detail in Chapter 4.

Bautista-Ballester, J., Vergés-Llahí, J., Puig, D.: Weighting Video Information into a Multikernel SVM for Human Action Recognition, In *Proceedings of the 8th International Conference on Machine Vision*, Barcelona, Spain, November 2015.

In order to increase the robustness of the recognition of actions in more challenging situations, we proposed an approach that was able to weight different sources of information relevant to discriminate actions, namely, the space and temporal features that describe the motion. Thus, the main contribution of this paper was the fusion and discrimination of new information sources for performed actions and it is extended in Chapter 5.

Bautista-Ballester, J., Vergés-Llahí, J., Puig, D.,: Combining Contextual

and Modal Action Information into a Weighted Multikernel SVM for Human Action Recognition, In *Proceedings of the 11th International conference in Computer Vision Theory and Applications*, Rome, Italy, March 2016.

In this paper we took advantage of the structure proposed in Chapter 5 in two ways: firstly, by adding data that was not strictly a descriptor of motion but modal or contextual information obtained by segmenting the region where the action took place in three space dimensions and describing the tool employed in the action, which was a new way of using multikernel SVM. Secondly, to determine which channel had more non-redundant information by weighting each information source.

Bautista-Ballester, J., Vergés-Llahí, J., Puig, D.,: Action Classification Using Multiple Sensor Information Sources fused into a Multikernel SVM Framework, *IET Image Processing*, Submitted manuscript.

This work presented an early version of the experiments presented in Chapter 5 including a time analysis of the approach.

Bautista-Ballester, J., Vergés-Llahí, J., Puig, D.,: Action Recognition Through a BoW Representation of Significant Action Objects, *IET Electronic Letters*, Submitted manuscript.

This work presented an early version of the experiments presented in Chapter 4.

Bautista-Ballester, J., Puig, D.,: Improving Action Classification with an Incremental Learning approach from Visual and Depth Sensors Using a Weighted Multikernel Support Vector Machine, *Robotics and Autonomous Systems*, Submitted manuscript.

This paper presented an Incremental Weighted Contextual and Modal MultiKerneL Support Vector Machine (IWCMMKL-SVM) approach for improving human action recognition. Basically, the work proposed an approach that allowed the incorporation of new training data to the classifier without having to train it

again from batch and without loss of performance. All the details of this work are presented in Chapter 6.

Lopes, C. M., Graça, J., Sastre, J.; Reyes, M., Bautista-Ballester, J., Guzmán, R., Braga, R., Monteiro, A., Pinto, P. A.: Estimativa automática da produção de uvas utilizando o robô VINBOT - Resultados preliminares com a casta Viosinho., In *10 Simpsio de Vitivinicultura do Alentejo*, Évora, Portugal, May 2016.

In order to promote the dissemination of the VinBot project (Appendix C), the consortium of the project has submitted a paper to be published in the proceedings. The work includes only a few results from the ground truth of Viosinho variety and image analysis from 2015 gathered data.

Lopes, C. M., Graça, J., Sastre, J.; Reyes, M., Bautista-Ballester, J., Guzmán, R., Braga, R., Monteiro, A., Pinto, P. A.: Vineyard yield estimation by VINBOT robot - preliminary results with the white variety Viosinho., In *XI International Terroir Congress*, Willamette Valley, Oregon July 2016, Submitted manuscript.

In this paper we presented and discussed the relationships between actual and estimated yield computed using the surface occupied by the grape clusters in the images. This paper had the purpose of promote and disseminate Vinbot project (Appendix C).

Appendix C

Real Applications

"In the modern world of business, it is useless to be a creative, original thinker unless you can also sell what you create."

- David Ogilvy,

Due to the Industrial character of this dissertation, I have had the chance to work in two real applications, VinBot and RoboHow. The former has been carried out at Ateknea Solutions Catalonia during the whole period of the doctoral program. The latter was carried out in the Learning Algorithms and Systems Laboratory (LASA) laboratory from the Ecole Politechnique Federale de Lausanne (EPFL) in Lausanne and my contribution to the project took place over a period of three months.

VinBot is an all-terrain autonomous mobile robot with a set of sensors capable of capturing and analyzing vineyard images and 3D data by means of cloud computing applications, to determine the yield of vineyards and to share this information with the winegrowers.

RoboHow aims at enabling robots to competently perform everyday human-scale manipulation activities - both in human working and living environments. In order to achieve this goal, RoboHow pursues a knowledge-enabled and plan-based approach to robot programming and control. The vision of the project is that of a cognitive robot that autonomously performs complex everyday manipulation tasks and extends its repertoire of such tasks by acquiring new skills using web-enabled and experience-based learning as well as by observing humans.

C.1 VinBot

VinBot responds to a need to boost the quality of European wines by implementing Precision Viticulture (PV) to estimate the yield (amount of fruit per square meter of vine area: kg/m2). The VinBot project aims to tackle these challenges through the development of an autonomous mobile robot enhanced with cloud computing applications to automatically acquire, process and present comprehensive and precise yield information to winegrowers and associations in the form of a web-based map. Through their collective expertise in viticulture, the winegrowers and associations could then coordinate crucial yield management techniques to improve efficiency and wine quality, per their commercial strategies.

C.1.1 Project Consortium

Due to the European character of the project, the consortium guarantees complementary and synergistic business interests, ensuring a quick and dynamic route to the market for the technology. All members are fully committed and motivated to work together to ensure the success of the project. The consortium is geographically very representative and includes four major wine producing EU Member States (Portugal, Spain, Italy and Hungary), and Romania.

Figure C.1: Consortium of the VinBot project.

The consortium includes wine sector SME Associations (SME-AGs) from four main EU wine producing countries: Portugal, Spain, Italy and Hungary, who altogether are responsible for almost 40% of the total wine produced in the EU. The role of the SME-AGs GRANJA, PROVIR, DALFONS and Orgoványi Gazdaszövetkezet are centered on i) supporting the R& D efforts, ii) training their associated SMEs, iii) disseminating the project at transnational level and iv) leading all efforts on the protection and exploitation of the results.

They work to expand the knowledge base of large communities of SMEs through comprehensive training courses to their SME members, complemented by demonstration activities to the management of their members, as well as to other SME Associations and key players in the viticulture industry across Europe. During the project they also have served as an invaluable connection between the RTD performers and industry by mobilizing their SME members to define the industrial specifications for the technology, guiding the R& D performers in their tasks and testing and validating the developed prototypes when they became available.

The Core Group of SMEs: The main role of the core group of SMEs selected by the SME-AGs is to ensure that the results of the project can be used by a large number of SMEs. This is achieved by taking an active role in validating the technology developed, advising and assisting other participants, as well as the taking-up, training and disseminating activities. They benefit from early access and preferential use of the results.

On one hand, the technological SME Agri-Ciência and ASSIST are responsible for the technology push, and are fully capable of developing, and commercializing the commercial VinBot platform. On the other hand, market pull will be created by end-user SME vineyards GRANJA, PROVIR, DALFONS and Orgoványi Gazdaszövetkezet. These SMEs fit the model European winegrower that composes the core of the target winegrower sector. As such, their main role is to contribute first-hand information of their daily activities and needs to the RTD performers to ensure the proposed technology is fully in line with industry expectations and realities. They also carry out exhaustive testing of the developed prototype in

their vineyards and interact with the rest of the consortium in communicating any problems, suggestions, required improvements, etc. Similar to the remainder of the core group of SMEs, they play an active and committed role in Training and Dissemination Activities.

The RTD Performers: ATEKNEA, ISA and ROBOTNIK have been selected on the basis of their expertise, resources and complementary interests. Aside from its project management expertise, ATEKNEA has extensive experience in the development of industrial solutions based on analogue and digital electronics, integrated analytic systems, physical instrumentation, wireless communications, artificial intelligence, machine vision, sensing solutions and IT management platforms. They are in charge of the development of the computer vision and image processing systems, and integrate the vinbot components.

ROBOTNIK has extensive experience in the development of service robotics for different sectors, such as agriculture, search and rescue, military, civil security, industrial inspection and other sectors. They are in charge of the design of the robot and software that controls it from a cloud service, allowing it to navigate the fields and map the vines' positions accurately.

ISA has been selected for its extensive expertise in precision viticulture and the use of methodologies for analysis in the agricultural sector, namely for the assessment of stress factors, diseases and constituents of plants and fruits alike.

These three RTD performers perfectly cover the technical needs of the project and have counseled the SME-AGs during the training activities.

C.1.2 Problem Statement

The goal of the project is to build and program an autonomous robotic platform with several mounted sensors to gather information about the state of the vineyards that could be used by growers to decide the tending policies to improve the quality of their product. This platform navigates autonomously along the fields, building maps, planning missions, learning the best trajectories to accomplish the missions, minimizing the risks and maximizing the success probability. During such missions

the robot gathers a series of data from heterogeneous sensors (visual, 3D, Normalized Difference Vegetation Index (NDVI), Global Positioning System (GPS)) which are processed and offered in a coherent and meaningful way to the final users (vine growers, producer associations). The key innovative mark of this project is that the processing of data are carried out partially off-board and accessed anywhere, that is, in a cloud service that drives, stores, analyzes, and shares the stream of data obtained by each robot. The scientific and technological goals to be achieved in this project are listed below.

The research goals in these fields are the procurement of data for building maps and navigation, the integration heterogeneous types of data, and the registration of visual data to precise geographical positions to anchor it to the vines. Specific algorithms have been developed to obtain the information required by growers from images of plants, canopy 3D structure, and vigor values.

The technological objectives are mainly related to the main elements that compose the system, i.e., autonomous mobile robot, sensory system, computer vision, and cloud service. The first pair is physical and mechanical elements (hardware), while the second pair corresponds to software and telecommunication capacities.

The specific goals that pertain to this thesis can be summarized as follows:

- Sensory system: the purpose of the robot is to gather precise information from sensors that would allow the growers to improve their tending policies. Therefore, the robotic platform incorporates several types of sensors, such as color cameras, Near InfraRed (NIR) cameras, 3D range finders and Global Positioning System (GPS). Research have been focused on the creation of suitable algorithms to extract the 3D structure of the canopy and estimations of the weight of grape clusters.
- Database generation: the gathering and publication of huge amounts of data by the robot have allowed not only the precision tending of vineyards, but also this data have provided us with material for research of other aspects of wine culture.
- Computer vision: the development and integration of computer vision

algorithms that obtain information about leaves, canopy, grape clusters and vigor (NDVI). The software is able to process the images obtained by the imaging system in a flexible way, i.e., partially on-board and partially in a cloud service. The number of images required for such processing must be enough to cover the whole extension of the vine canopy for each single tree in a row, which depends on the robot speed, camera position and horizontal space covered by the image.

In order to achieve successful results from the VinBot project, the work to be carried out was previously organized into Work Packages (WPs). All WPs are classified differently, depending on their nature: Research, Technological Development and Innovation, Demonstration, Project Management and Other. What pertains to the specific work done for this dissertation is summarized as follows:

Firstly, implementation and integration. The objective of this WP is to specify, build, and fully integrate all components required in the process of capturing, storing and transmitting images, with the additional construction of a mechanical structure to house such elements to be mounted and connected to the robot equipment and software. This WP specifically deals with the elements required to do so in an effective way, namely, capture, illumination, storage, and transmission. The integration of all these elements with the structure of the robot and the tests of their operability are also taken into account in this WP.

Secondly, the development of the sensory system. The objective of this WP is to develop a computer vision system that perform the task of obtaining the images and 3D data from the sensors, transmitting and storing them into a web service on the cloud, and extracting and analyzing the information required to generate maps that allow the growers to take action in their vineyards. This vision system captures images of the vines with cameras mounted on the robot and also 3D range finder to measure the canopy. It is also capable of obtaining the visual information required to navigate, to build maps of the environment, and avoid obstacles. From the images obtained, the system manages their automatic processing in an off-board cloud service that extracts measures of the size of the canopy and the number, size,

and appearance of grape clusters. This data is later used to create maps of vegetative growth, vigor, and yield estimation.

C.1.3 Sensory System and the Acquisition of Data

This sections deals with the elements of the sensory head that allows the obtainment of the vineyard data. Also we describe the structure and dimensions of the head as well as the hardware and software elements that encompass it.

According to the previous specification of the vineyard structure and elements in the vines that must be characterized, the sensor head houses the following sensors:

- 3D Laser Range Finder (LRF): 1 unit situated at the sensor pole center to obtain the 3D data of the point clouds, i.e., (x,y,z).
- RGB+NIR Camera: 2 units used to obtain both RGB and NIR images of the canopy to compute the NDVI.
- Global Positioning System (GPS): To locate the position of the sensor head (independently of the robot). It provides data in National Marine Electronics Association (NMEA) format, which can be easily parsed. Information contained in the files includes Latitude, Longitude, Height, Horizontal and Vertical Deviations.
- Inertial Motion Unit (IMU): It provides 9 Degrees of Freedom (DOF) values corresponding to 3-axis gyro angles, compass, and accelerations of the sensor unit.
- Computer: Used to control the cameras and sensors, collect all the information, and send it afterwards to another computer via wire, WiFi, or a cloud service.
- Power Supply: For feeding the unit independently of the mobile platform
- Structure: It can be mounted on the mobile platform to house all the cameras, sensors, battery and computers independently. This structure can also stand alone on different types of platforms such as trolleys, garden charts or mobile robots.

The structure allows the height and configurations of the sensors to be changed, as well as the relative position of the sensor column with respect to the position of

the trees (closer/farther to/from the center of the line). This specific feature and all the components are visible in Figure C.2 Also, the sensor head is mechanically stable once mounted on a mobile platform, to avoid swaying or capsizing.

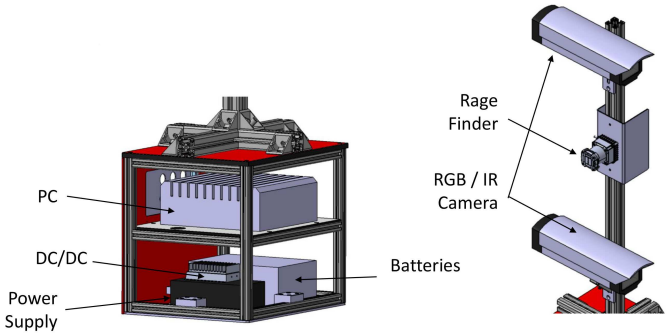


Figure C.2: Elements forming the sensor head unit.

C.1.3.1 Sensors

According to the previous specifications, the sensor unit must obtain information about the appearance and weight of grapes and the size of the canopy of vines, its structure and position. This information is obtained with cameras, a 3D range finder, and a GPS unit. Also, an IMU is incorporated to obtain more precise 3D points as explained later in this section. The cameras are a combination of RGB and NIR sensors. They make it possible to obtain an interesting set of images and also to compute the Normalized Difference Vegetation Index (NDVI), a standard measure of health, vigor and biomass of plants. The 3D range finder obtains 3D information in the form of clouds of points that can be processed later to compute evolving surfaces approximating the shape of the objects.

The camera selected for the job is a combination of color and NIR sensors which image the same scene by means of a prism in the optic side of the camera. This allows a single camera to provide two images, color and NIR, with exactly the same content. It can be understood as a low-cost spectral camera with for filters, RGB, and NIR (Figure C.3). Near InfraRed sensitivity is often used for traffic applications or surveillance/situation awareness, and can also be utilized in the identification of blemishes or defects not easily observed in the visible spectrum. NIR is also a

valuable tool because it provides a wider range in the wavelength spectrum beyond color and allows the computation of NDVI images, which is correlated to the vigor of the plant.

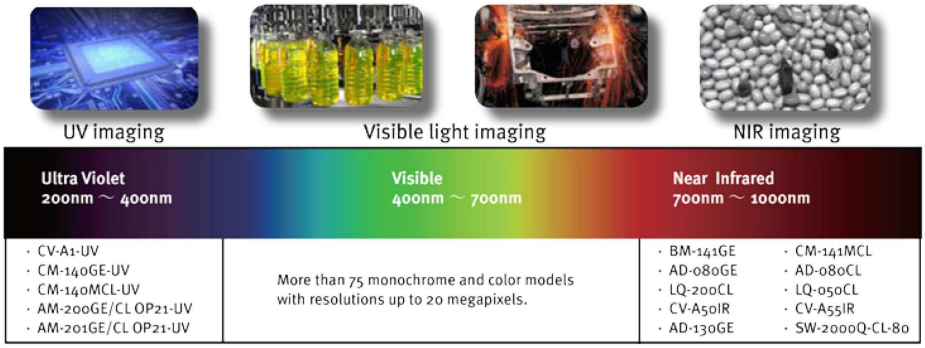


Figure C.3: Some of the JAI models classified according to their wave length range.

In order to study the development of the vegetation of plants we needed to extract the canopy structure, which is usually solved by obtaining 3D information, a solution that has been attempted in different field robotic applications. This data generates point clouds of the surface of the objects in the environment.

A popular choice for such types of applications is the URG series range finder sensor provided by Hokuyo Automatic. The Hokuyo URG-Series sensor is specially suited for outdoor application, with IP67 protection, accurate multiecho at distances of 30m and a view window of 270º.

We employ an EVK-7P evaluation kit to localize the robot throughout the field. U-blox 7 evaluation kits are compact, and their user-friendly interface and power supply make them ideally suited for use in laboratories, vehicles and outdoor locations. Furthermore, they can be used with a PDA or a notebook PC, making them perfect through all stages of design-in projects.

In addition to the GPS kit, a proper antenna is necessary to obtain a decent precision in the location measures. We use a L1 & L2 Glonass helical active antenna Maxtenna, which is designed for applications requiring greater accuracy than what L1 antennas alone can provide. The antenna is built from proprietary Maxtena Helicore® technology. This technology provides exceptional pattern control, polarization purity and high efficiency in a very compact form factor.

We use an IMU to capture a series of acceleration and gyroscope readings such as rotation angles that allow the inertial control of devices. In our case, we want to use such a unit to filter the likely sway of the VinBot while moving along the vineyards from the 3D point measurements obtained from the range finder. The unit mounted in the sensor system is a 9 DOF ArduIMU that includes three sensors, an ITG-3205 (triple-axis gyro), ADXL345 (triple-axis accelerometer), and HMC5883L (triple-axis magnetometer). The outputs of all sensors are processed by an on-board ATmega328 and output over a serial interface. This enables the 9 DOF ArduIMU to be used as a very powerful control mechanism for UAVs, autonomous vehicles and image stabilization systems.

C.1.3.2 Structure

The sensor unit is roughly divided into two parts: a pole in the upper part, where most of the sensors are attached, and a lower housing for the computer, communications, power supply and the rest of components. The idea is that the unit would be set on the mobile platform, but also can work autonomously. The present design only obtains data from one side of a vineyard line. The pole is made of aluminum IBM structural profile, which reduces the total weight. The box is also built from this material, and encased by a methacrylate board to keep the interior devices safe from the water, moisture, and dust outside. Vineyards are usually a dusty environment, but they can also be rainy and dump, especially during the periods of when the sensor unit is most likely to be used. Figure C.4 depicts the physical measurements of the sensor unit.

The sensor unit includes a 12.8V / 40Ah Polymer LiFePO4 rechargeable battery. Together with a 16V DC/DC converter, an internal power supply of 12V is provided for all the sensors in the unit. The sensor system also has a computing unit in the lower part of the body. It is a fanless industrial PC with computational power similar to a desktop. Despite the fact that VinBot is a project that focus on developing cloud-based services to process the data, in the very first steps of the project we decided to provide a powerful onboard PC to control the sensors, store, and process

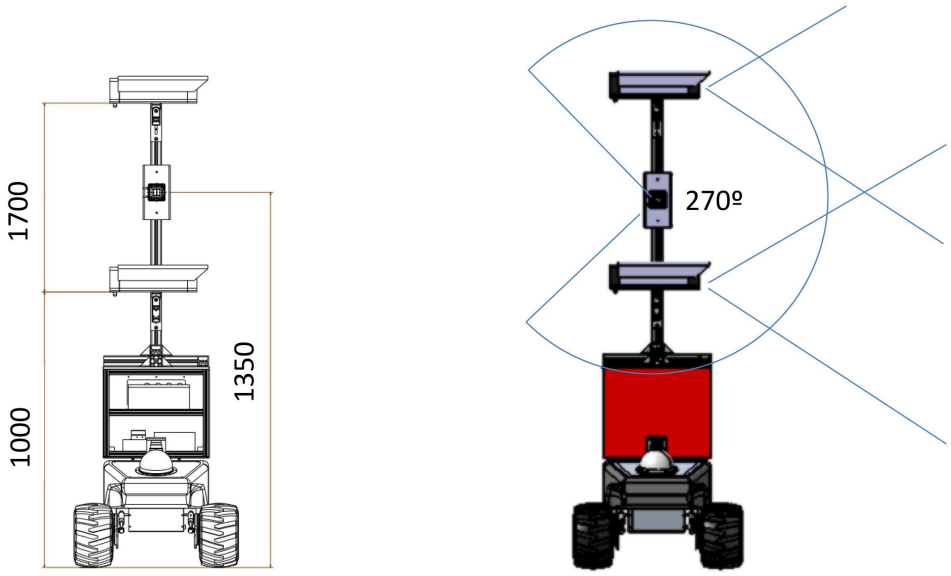


Figure C.4: Dimensions of the VinBot system expressed in millimeters.

data. Once these tasks are achieved on board, we move them to web-based ones and the PC requirements are reduced accordingly.

This PC incorporates cutting-edge 3rd generation i7 Quad-core processor and versatile I/O functions such as Gigabit Ethernet ports, and 3.0 USB ports. This computing power ensures video capture and analytic tasks running without failures. Also it has an ignition control that provides several intelligent control schemes to turn on/off the computer at the right time. This is important in field operations which are generally carried out without specialist intervention and no monitor to see the state of the computer.

With respect to the camera data transference, there are two ways of connecting the cameras to the computer, that is, using one switching hub and 1-port Ethernet Network Interface Controller (NIC) or each camera to an Ethernet port. The first configuration reduces the burden on the internal bus and CPU but can drastically limit the amount of data transmitted from the cameras or create delays in their transfer too long to be tolerated. The second option allows higher data transfer rates at the expense of overloading the CPU and the internal bus where the NIC ports are connected.

We experimented with both configurations and it was clear that if we wanted to

capture all the information coming from the cameras we needed a dedicated network port for each sensor, i.e. the second options is used at this moment. With the multi-port NIC, as represented in Figure C.5, each camera is connected to one port of the NIC. Each pair of connected cameras and the NIC construct one IP configuration and it is appropriate to use a persistent IP. Then, each camera can use the maximum 800Mbps bandwidth. However, the load for the internal bus, CPU and the applications become heavy, so this is why a powerful PC is required.

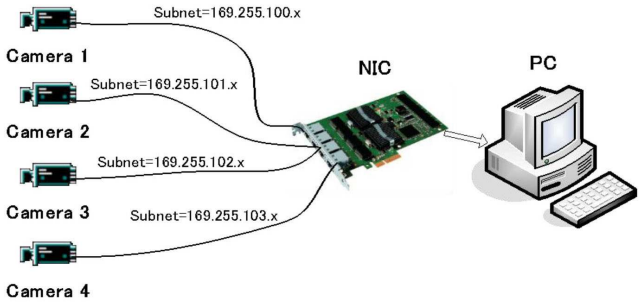


Figure C.5: Connecting a camera using an n-port NIC.

C.1.3.3 Software

The sensor unit have a heterogeneous variety of sensors. Cameras, 3D range finders, GPS, and IMU, to be specific. An API is required to allow us to achieve two objectives: using the SDKs provided by each device maker and creating a common API for all the sensors.

A middleware is an abstraction layer of software that resides between the Operating System (OS) running the computer and the software applications. It is usually designed to manage the heterogeneity of hardware existing in a certain system, to improve the quality of software applications, to simplify software design, and to reduce development costs. The idea behind the middleware is that a developer only needs to build the algorithm as a single component. This component can be combined afterwards and integrated with other existing components. Furthermore, whenever the component is modified, the programmer only need to replace the old one with the new one, which improves testing efficiency. This architecture is adopted

in VinBot and it is summarized as in Figure C.6.

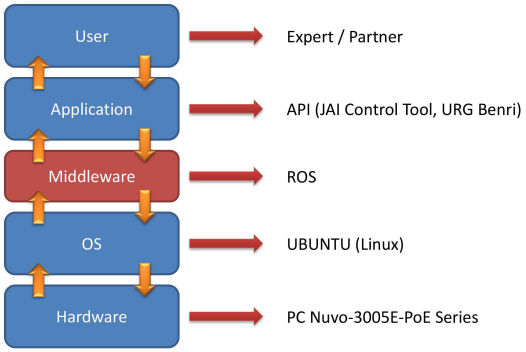


Figure C.6: VinBot software architecture.

Based on Robot Operating System (ROS), we can see all the sensors in the same screen at the same time in just a single click. ROS provides a user interface called Rviz in which the model can be seen on screen and the data captured is represented in an intuitive way (Figure C.7). Furthermore, the captured data can be seen over the same screen and we can determine if the capture is going to be well recorded in a glance.

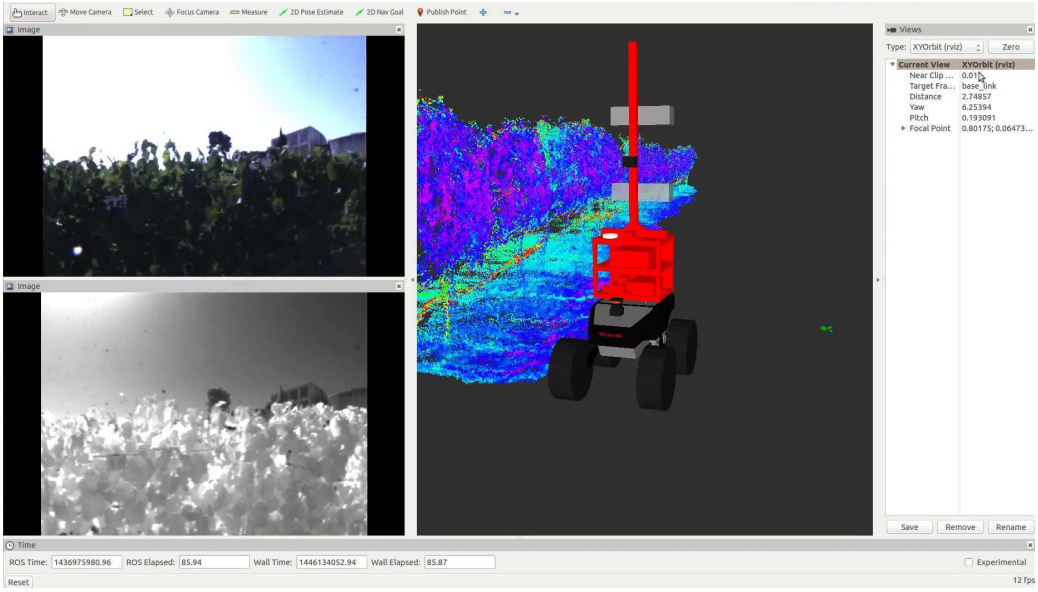


Figure C.7: API based on Rviz ROS visualizer.

Bag files recorded are split into three basic files according to the type of information saved:

- Vinbot_ cameras.bag : compressed NIR and RGB camera topics
- Vinbot_ vinbotsys.bag : compressed Range Finder, GPS, IMU topics
- Vinbot_ diag.bag : compressed diagnosis topics

When plannig to gather some data, the procedure followed by the operator has to follow this sequence:

- Click on vinbot icon
- Enter the variety in an input screen
- A recording session is generated automatically, saved in a folder with the same name and with a timestamp, all threads are executed, and the recording starts automatically.

It is necessary to push the sensor head through the vineyard rows to collect data from the right side of the row. The procedure is described in Figure C.8.

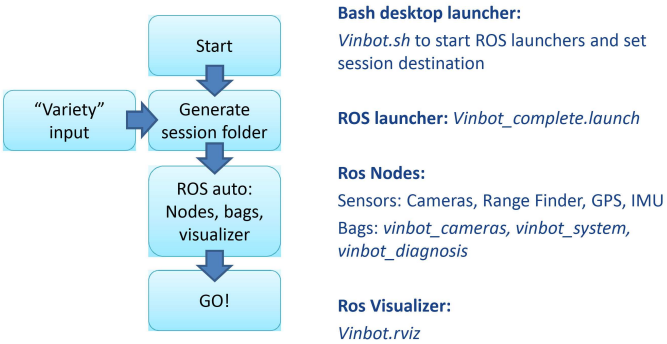


Figure C.8: General procedure when gathering data with VinBot head v.2.

C.1.4 Preliminary Results

In the first task, extraction of grape features, we developed the algorithms required to obtain the shape, size, and appearance features that describe the grapes using images gathered by the mobile robot as input. This information was obtained at several stages of the growing process and was employed to establish yield estimation of the plots as well as the maturing state, required in differential harvesting procedures. In traditional vineyard yield estimation, the crop components that are measured to derive a final estimate are (1) number of clusters per vine, (2) number of berries per cluster, and (3) berry size grape clusters.

Our approach consists of determining the regions of the images that contain grapes forming clusters, using different measures of color and NIR. Yield estimations are obtained as number of clusters and the total size of the image regions that present grapes. This is based on previous approaches that find a correspondence between the image area with clusters and the production of grapes in weight (Tardaguila et al., 2012).

In the second task, extraction of canopy features, the canopy architecture and density determine the microclimate and performance of the vine that significantly impact fruit maturation, composition and date of harvest. Measuring canopy size during the growing season can give the vineyard manager information to make necessary manipulations to their vines to optimize their crop. Structure and 3D data are obtained from sensors and used to create descriptions of the shape of vineyards. 3D shape information of vines is required to determine several features involved in the management of canopies, like size, volume and exposed area.

Once the 3D data is scanned using laser ranger scanners, a registered 3D model is generated and the measurements of the canopy size can be obtained. Additionally, 3D information allow us to associate precise spatial coordinates with this data. We also extract information relative to the exposed leaf area from the RGB+NIR images. These images provide a different perspective on the structure of the canopy and can be used alternatively to describe the porosity of the vineyard. By comparing this information from different periods of the vegetative growing, experts are able to determine the course of the growing and to create maps of growth.

C.1.4.1 Grape Features

This section provides a detailed description of the methodologies developed to detect and determine the number and size of the grape clusters using the images obtained by the RGB+NIR cameras mounted on the robot.

One of the most important parts of this work consists in computing and using the NDVI from the RGB+NIR cameras that are mounted on the sensory head. NDVI is a simple indicator that can be used to analyze remote sensing measurements,

typically but not necessarily from a space platform, and assess whether the target being observed contains live green vegetation or not.

Figure C.9: NDVI is calculated from the visible and near-infrared light reflected by vegetation. Healthy vegetation (left) absorbs most of the visible light that hits it, and reflects a large portion of the near-infrared light. Unhealthy or sparse vegetation (right) reflects more visible light and less near-infrared light.

The advantage of using NDVI values is that these indices are normalized within the interval [-1..1] and we can segment the images into different classes based on their values. In fact, NDVI is a perfect way to detect leaves in the image, as can be seen in Figure C.9.

The methodology implemented consists of a two stage approach. First, we perform a coarse classification of the different elements present in the images. This is based on several features obtained from the color images. The candidate regions are used directly to compute the characteristics related to the grape production (yield) of the vineyard. Nevertheless, we use these results to create a database of images that are employed to train the Object Recognition based on BoW (ORBoW) engine that is used in the recognition stage (test) to find the regions in the image that correspond to each class used for training.

In the images we are mainly interested in finding pixels that belonged to grapes. However, in a more general way, we define a list of classes that were likely present in all images picturing vines. These classes are: sky, grapes, leaves, soil, and other. Sky encompasses everything that appeared in the sky, from clouds to different types of sky (sunny, cloudy, etc.). Grapes correspond to the regions in the image that contain clusters of grape. Different varieties have changes in color, shape and size.

Leaves only contain green leaves. Soil is a class for the ground and other dry parts in the image, such as brunches, leaves, and grass. The rest of the pixels that are not classified into any of the previous classes are labeled as other.

The following stage is to automatically obtain samples to train the ORBoW engine to detect certain classes in images. Samples corresponding to a given class are stored as separate images cropped from the original in a separated folder. Also a file with the list of all the file names obtained is generated in order to perform posterior batch processes. This list can be modified manually later on in order to reduce the size of the set of samples, or leave some outliers out of the list.

The last step in this process consist in the extraction of features from the samples obtained in the previous section. For the description of these samples we use the traditional Bag of Words (BoW) image descriptors employing SIFT and SURF point descriptors. The extraction of features is performed as a batch process that read a list of file names that correspond to the samples, and the extraction is performed on each image from the list.

We compute different BoW image features using SIFT and SURF descriptors computed in a group of points. Despite the fact that there exist many algorithms for detection of Interest Points (IPs) as seen in Chapter 3, we decided to use a fixed dense grid of points on the image patch. This way, the number of features per sample is always constant and can be easily controlled. The idea is to compute a description of the distribution of these descriptors per image patch. This is obtained by computing a BoW per patch, which consists in determining the relative occurrence of a series of features that are used as reference (vocabulary). Therefore, an image is described by the histogram of the visual words in the vocabulary for each codebook, the one based in SIFT and the other based in SURF. As seen in Chapter 3, the computation of the vocabulary is a key step since it determines the description of a set of images.

In our case, the whole process rely on the sets used for training the machine learning algorithms that are employed later for detecting the regions in the image with the corresponding class. We employ the same sets for the computation of the vocabulary, which is obtained by clustering (using a k-Means algorithm) a randomly

selected set of features into a given number of clusters. As verified in previous experiments (Chapter 3), this number determines the dimension of the BoW image descriptor and, in case the set of features to cluster exceeds a manageable number, an alternative solution consists in randomly selecting a small number of vocabulary words from the set. Once a vocabulary is obtained, the set of features per image, be it for training or testing, is transformed into a BoW, that is appended to a file with all the features describing a given set of samples. These values are used later for training classes and computing matched in images.

Figure C.10: Region classification results obtained by using the ORBoW approach, combining SIFT and SURF local descriptors to build the codebook.

In order to find a certain structure in the images corresponding to classes such as grapes or leaves, we run the ORBoW engine developed in this dissertation starting from the simplest version of it, which encompasses a multikernel SVM for SIFT and SURF based codebooks. We classify only one class against the rest. For this purpose, given a certain class, i.e., grape, a training set is created with positive samples (grapes) and negative samples (leaves). The size of this training set is selected by the corresponding parameters. Also a test set required to check the outcome of the training process is generated with the remaining available samples.

Then during testing, it is important to construct the test feature vectors in exactly the same way, scaling the test inputs using the saved means and standard deviations, prior to sending them to the SVM classifier. Figure C.10 represents the results of grape class classification.

One of the most challenging points for this training is the fact that the robot

creates a set of images for each row of the variety vineyard, which includes an increasing number of database sessions for the whole season and for all varieties. This means that every time the robot collect new data a batch training is necessary. However, with the approach proposed in Chapter 6 of this dissertation we are able to incrementally learn new features of the vineyard, providing the classifier with more data avoiding a batch learning after each recorded session.

C.1.4.2 Canopy Features

Canopy architecture and density determines the microclimate and performance of the vine which significantly impacts fruit maturation, composition and date of harvest. Measuring canopy size during the growing season can give the vineyard manager information to make necessary manipulations to their vines to optimize their crop.

This section deals with the computation of the features that describe the structure of the canopy. Our main interest is to measure the size of the canopy in terms of area and volume. Although for this purpose we employ the data obtained from the 3D range finder, we also want to investigate the possibility of extracting relative measurements of leaf and surface areas using the RGB+NIR images. This is a totally different sort of data that also provides an idea of the porosity of the vineyards in similar terms as the porosity measures of tree canopies that are computed from beneath the tree.

The main step consists on employing the segmentation results obtained in the previous section to compute the indices. In FigureC.20 we can see the results of the segmentation based on the NDVI values. These images are generated by applying different masks (segmentation result) on the (corrected) RGB image. The masks correspond to the pixels that belong strictly to the leaves (top-left), the holes in the leaf region (top-right), the canopy consisting of leaves and their holes (bottom-left), and the rest of the image that does not belong to the canopy nor to its holes (bottom-right).

In frontal images of vines we consider the canopy as the image that encompasses the whole set of leaves. Leaf Area Index (LAI) and porosity values are computed

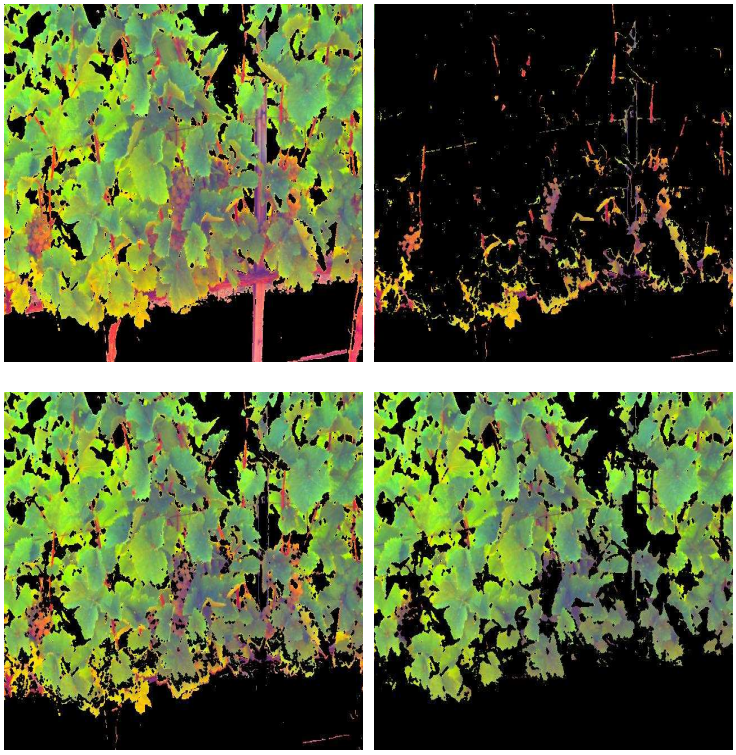


Figure C.11: Resulting segmentation based on NDVI values. The top row shows the corrected RGB image considering all the positive NDVI values (left) and all the negative NDVI values (right). The bottom row shows the corrected color image with all the canopy (left) and only the leaves (right).

based on different relation between the canopy, the leaves, and the holes in the canopy obtained from these images.

C.1.4.3 Canopy 3D Features

3D shape information of vines is required to determine several features involved in the management of canopies, like size, volume and exposed area. By comparing this information from different periods of the vegetative growing the algorithm is able to determine the course of the growing and create maps of the growth.

We use a 3D laser sensor (Hokuyo) in order to obtain the 3D information of the canopy, such as structure and its descriptors. The laser sensor is mounted on the head of the robot. In order to register the whole line of the canopy, the laser must move along the vine and take measurements for each fraction of time, in this case, at a frequency of 40Hz. The faster the robot moves the fewer the number of measures

we have. It is necessary to take this into account when detecting the thinner elements of the plant.

The point cloud obtained by the 3D range finder is a set of distances relative to the center of the device. This saved data must be threaded in order to convert it to a set of Cartesian (x,y,z) points. Regarding the vertical measurements of the canopy, the range of the laser is 270ºand the device takes 1080 points for each spin done over its own axis. This means that the angular resolution is 0.25º. This can be related to distance when we consider the distance between the device and the obstacle. For example, if the robot goes through the vineyard along a line $1m$ away from the canopy, then the resolution we can obtain in the vertical axis is $1000 \cdot tan(0.25) = 4.3mm$. From this value, we can ensure that we are not able to measure shots thinner than $4.3mm$ when we are capturing data $1m$ away from the canopy.

With respect to the z axis, which is considered here as the axis along the vineyard, the resolution basically depends on the robot's speed. Considering a velocity of $3m/s$, with a frequency of the range finder of $40Hz$, we have 40/313 measures per meter. This means that the resolution in this direction when moving at 3 m/s is $1000/13 = 76.9mm$. The slower the robot goes, the better the resolution is.

When all data points are converted to Cartesian 3D points, we are able to see something like FigureC.12, which is a 3D reconstruction of all the points represented as a cloud.

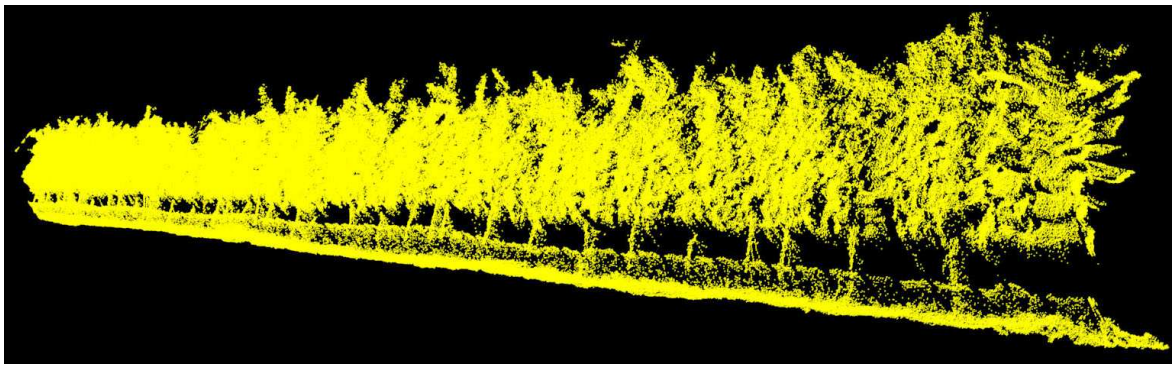


Figure C.12: Representation of the point clouds obtained with the 3D range finder.

The vine growers need to make necessary manipulations to their vines to optimize

their crop. In order to do that, they use some parameter estimators such as the LAI. Two techniques are proposed to estimate this LAI. One option is to use the PCL library, a standard C++ implemented library which includes specific functions to deal with point clouds. Using these functions, we are able to calculate a Delaunay triangulation between points of the cloud, and then, estimate the area of the canopy by summing up these triangle areas. The second option, proposed by us, is based on the calculation of convex hulls for cross sectional areas and volumes. With this approach we calculate either the area of a slice captured by the sensor and integrate it along the row of the vine or the whole canopy volume.

We tested both options. The first method had a high precision, but a high computation cost. The tests performed indicated that the approach was not feasible because of the number of holes remaining during the triangulation process. We needed at least 90% of the canopy fulfilled by triangles, and we had less than 5%. This was due to the irregular surface of the canopy. However, the PCL functions worked perfectly when the surface was smooth, as for example with a sphere or a square. The Second method integrates areas along the z axis, meaning that we have some estimation between measures. We expected it to have a low precision, but when the approach was tested, we found that the precision was acceptable. Also considering the computational cost of the approach, this way is much less expensive than using PCL library functions.

Then, we consider that every canopy is separated by 1 meter and we keep the points that represents each one. We calculate the convex hull for each section captured for this canopy and integrate the values along all measurements. We realized that outliers could affect the results too much, and this is one of the reasons why we did not use a convex hull for the whole volume of the canopy. By using the cross sectional area calculation process, if an outlier exists it only affects the calculation of this area. This is evident in the holes that exist in the canopy. They are inside the canopy area, but they must be considered outliers because they are far from their neighbors. Canopy number 3 of the row named 9E and 10W (both sides) of the Viosinho variety used for tests is shown in Figure C.13. First, a 3D

representation of the two halves of the row is visible. Below, one cross Sectional Area is shown with the convex hull represented in green. This figure shows how the outliers affect the calculation of the sectional area. In this case, these outliers are expected to be holes, but it has to be verified with the information provided by the RGB cameras.

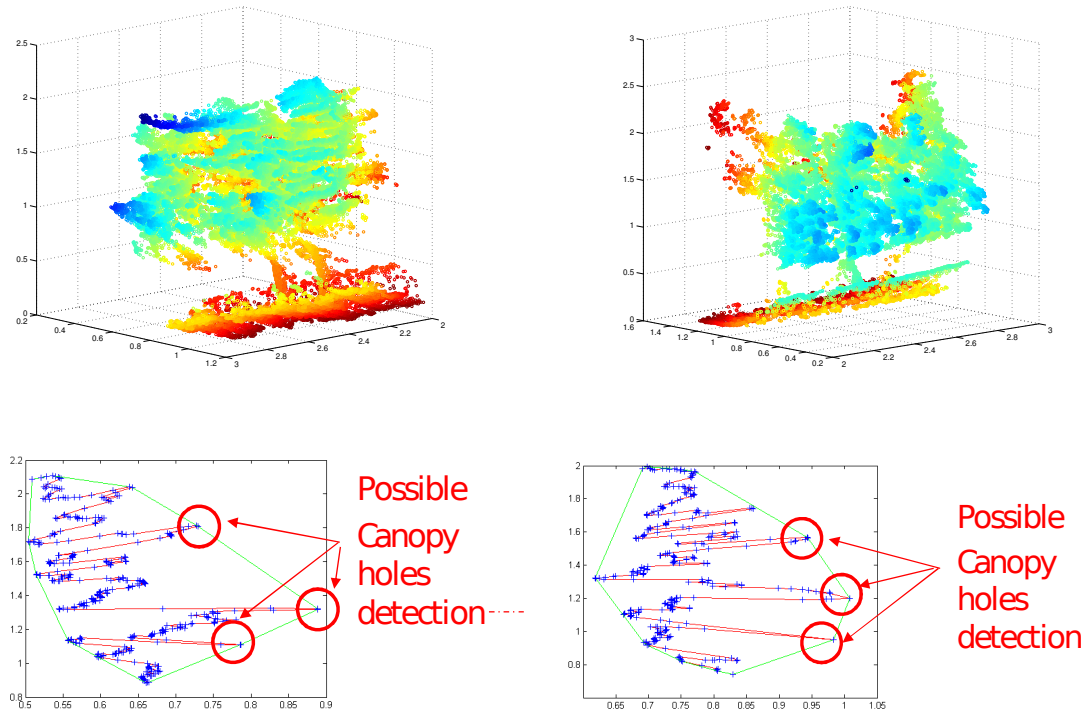


Figure C.13: Two sides of a single vine: 9E/10W (left/right images) n°3 and slices of the vine using Convex Hulls for the Cross Sectional Area.

The parameters that can be estimated with the 3D laser are shown in Figure C.14 and are as follows: height (H), width (W), volume (V) of the canopy and the cross Sectional Area (SA). Indirectly, we are able to also estimate LAI and TAI. This last parameter is used by Arnó et al. (2013) in order to estimate the LAI.

In the end, volume estimation is done and compared with the ground truth previously created. In Figure C.14 we can see the two estimations done with convex hull approach and the ground truth of Viosinho row 9E+10W. The difference between the two estimations are that we can compute the convex hull of the whole volume of the canopy, which is represented in red, or we use the cross Sectional Area method, represented in green. We see that the cross section method underestimates the

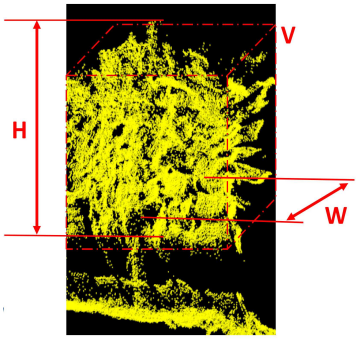


Figure C.14: Canopy features to be estimated. Height (H), width (W) and volume (V) of the canopy can be estimated from 3D points, as well as the cross Sectional Area (SA).

volume and the best approximation is due to the convex hull over the whole canopy. This is because the cross sectional method needs a good filtration of the 3D points, and in our case, we are still not controlling this filtration automatically. Thus, we set a quite restrictive cubic threshold and this obviously affects the volume calculation. Nevertheless, we know that the ground truth is overestimating the real volume due to its calculation process as the volume of a rectangular prism. Hence, we cannot consider this underestimation as a big error. On the other side, calculating the volume of the canopy as a whole creates a good approximation, but we can see the unpredictable effect of the outliers in the canopies 1, 18, 19, 22, 23, where the estimation is quite bad, either under or over estimating.

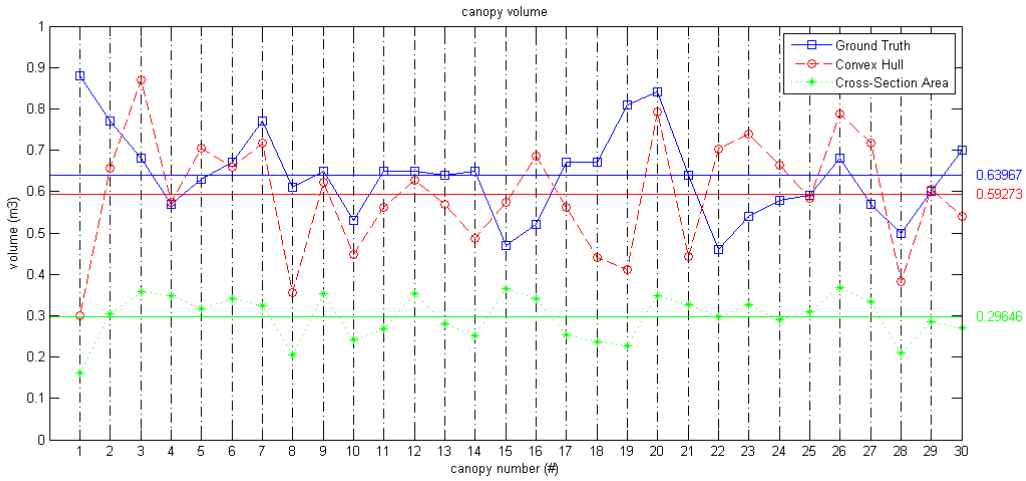


Figure C.15: Volume estimation by convex hull approach and the ground truth for the whole row 9E+10W.

C.1.5 Conclusions and Future Work

The first version of the sensor unit has been tested and widely used throughout the same vineyard for an entire season. At the present moment, we have most of the main components integrated (cameras, LRF, PC, power supply, GPS, IMU). The images in Figure C.16 illustrate the robot construction and the actual stage.

Figure C.16: Vinbot unit phases. Preliminary tested sensor unit on the left, first version in the middle and the fully integrated version on the right.

We showed the technical specifications of the VinBot system focusing on the sensors required to meet the specifications necessary to obtain information from the vineyards.

First we have taken into account the description of the structure of the vineyards. The size and system of growing of the vineyards have a direct effect on the dimensions of the sensory head. Also, the sort of measurements to be performed on the vines and grapes determine the nature of sensors. We have specified such requirements and also enumerated the actual sensors that are used in the construction of the sensor head. Additional elements that have been described in this chapter are the mechanical structure that have been designed and built to hold the sensors, the computational unit, communications, and power supply of the head. Also a description of the existing and software under development required to control the head have been described.

We have considered the tasks of developing methodologies for extracting features

from images and 3D data that provide information about the yield and vigor of the vineyards. First, we have considered the problem of localizing grape clusters in images. Our approach consists of employing the information provided by RGB+NIR to detect the regions in the image that plausibly contain grapes by running an ORBoW engine. Color has been used for this purpose, combined with the computation of NDVI.

Secondly, we have performed computations on the vegetative characterization of the canopy. Primarily, we have taken advantage of the NDVI images obtained to allow the computation of a series of indices that describe the amount of leaves exposed to the sun and the canopy porosity. Also, we have employed the 3D information obtained by the laser range finder to compute 3D structural parameters such as height, width and volume.

We also have a method to estimate volume of the canopy, which has been tested and contrasted to other methods, such as PCL, and we are quite ambitious to get a better performance if we can do a better filtering of the points. The results can be improved by fusing both rows making use of the GPS information. Using global coordinates, the cloud point would include both sides of the canopy, facilitating the filtration of outliers and limiting their number.

The work of this project is ongoing and more data must be obtained from the sessions in the ground truth database to compare these features to those obtained by hand and determine the degree of correspondences.

Since the project is still ongoing, the future tasks consist of several points.

- Despite the fact that the features make reference to phytophysical features of a vineyard that are already known to be correlated to yield and vegetative characteristics of vineyards, we must study more deeply the correlations between the measurements obtained by hand and the results from the automatic process of image.
- Further experimentation with an increasing number of images and varieties. It would be interesting to reduce the required manual setting of certain parameters that are variety dependent, and see how such parameters evolve

with different varieties and periods of time.

- Using 3D data to calculate LAI and Porosity indexes.
- Match resulting holes predicted by ORBoW engine in RGB images and the ones predicted with the 3D point clouds.
- Usage of the multimodal representation of the vineyard to improve ORBoW engine performance.
- Integrate the incremental approach proposed in Chapter 6 to progressively incorporate new data acquired without having to learn from batch.

C.1.6 Dissemination

The objective of the dissemination is to ensure that non-confidential information about the VinBot project and its results are disseminated to a wide and relevant audience to extend the impact of the project results. All consortium members have been actively involved in carrying out dissemination activities through their own contacts networks.

C.1.6.1 Website

During May 2014 the website for the VinBot project was launched (www.vinbot.eu). As illustrated in Figure C.17, the design followed the visual identity created through the logo, the leaflet and the poster. The content was initially proposed by the Coordinator and the rest of the partners provided their feedback to improve it.

The website presents the general information about the robot avoiding any confidential disclosure. It includes a section about the wine industry in Spain, Portugal and Hungary (the countries where the SME-AGs are located and important producers in Europe). It also includes a section devoted to the explanation of the main technologies that are the base of VinBot.

It includes a description of the partners in the Technology Validation section, since the partners are the developers and the validators of VinBot.

In the NEWS section, partners can follow the general progress of the project and can download the public documents generated during the project.

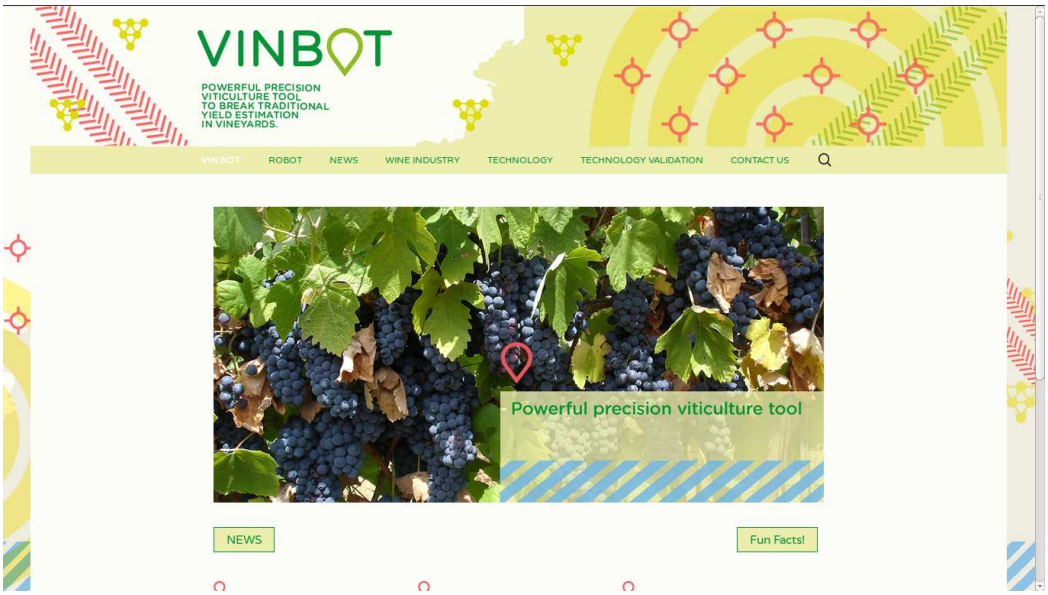


Figure C.17: VinBot website developed for dissemination purposes.

C.1.6.2 BTA Expo in Barcelona

ATEKNEA was present at the fair Barcelona Tecnologías de la Alimentación (BTA) from 21st to 24th April 2015. At its 15th edition brought together more than 35,000 visitors fascinated by the latest technology, machinery and ingredients for food. The team of ATEKNEA (Dr. Jaume Vergés and Jordi Bautista), Robotnik (Roberto Guzmán and Daniel Carbonell) and ISA (Carlos Lopes and Ricardo Braga) showed the VinBot prototype. Figure C.18 shows the team at the Ateknea's stand and the robot 3D visualization of it during the demo. Certainly, being able to see the robot attracted many onlookers, who took an active interest in the commercialization of technology once the validation phase finish.

The main message sent was to show the new technology being developed for the modern viticulture with the support of the EU. Visitors were told about the autonomous mobile robot capable of measuring the state of the vineyards and sending these measures to an intelligent central system that helps the winemaker to take contrasted decisions. The message and the audiences addressed (industry and civil society) were in line with those defined by consortium partners. A promotional video was made from the participation at the event.

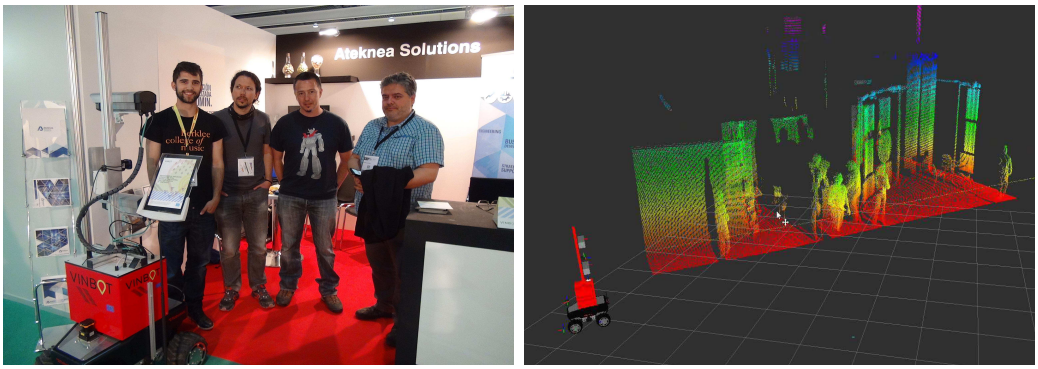


Figure C.18: VinBot team during BTA exposition and a 3D point cloud generation of the stand.

https://youtu.be/YZcIiCsbAjI

C.1.6.3 Successful VinBot demonstration session carried out in Lisbon.

The demonstration of the VinBot project took place on Thursday 16th of July, 2015, at ISA facilities in Lisbon. The demonstration showed the autonomous functioning of VinBot to the general public.

Figure C.19: VinBot demonstration and the robot operation during the demonstration held in Lisbon, 2015.

As can be seen in Figure C.19, the event was well received among both the public and the media, such as in the Portuguese daily PUBLICO and French LAVIGNE.

http://www.publico.pt/tecnologia/noticia/vinbot-o-robo-todooterreno-para-a-vinha-1702366

http://www.lavigne-mag.fr/actualites/robot-vinbot-estime-la-vigueur-et-les-

rendements-artFa-107027.html

A promotional video was realized from this demo and it is publicly accessible.

https://youtu.be/rVRXQvHoilw

C.2 RoboHow

Kinematic and video demonstrations from robot-assisted procedures can be used for LfD, developing finite state machines, assessing surgical skills, and calibrating. Learning tasks are often multi-step procedures that have complex interactions with the environment, and as a result, demonstrations are noisy and may contain superfluous or repeated actions. Temporal segmentation of the demonstrations (Figure C.20) into meaningful contiguous sections facilitates local learning from demonstrations and salvaging good local segments from inconsistent demonstrations.

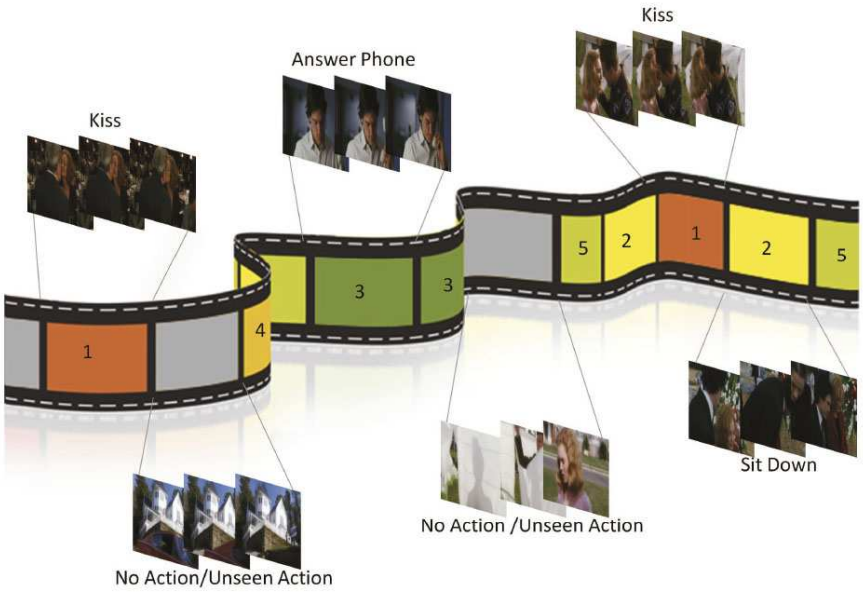


Figure C.20: Segmentation of a recorded task into meaningful contiguous sections. Our method can handle multiple action classes, including the null class of idle activities. (reprinted from Hoai et al. (2011), ©IEEE).

There is a large and growing corpus of kinematic and video recordings that can potentially facilitate human training and the automation of subtasks. For these recordings, manual segmentation is prone to error and impractical for large databases. A number of recent studies have also attempted to segment human motions from videos, either with supervised or unsupervised models. Robot data (Figure C.21) is used in LfD to obtain the control policies that allows the robot to perform the task, but no information from either the environment or the objects or

tools manipulated are incorporated into the system. Figure C.22 shows the features that visual information can contribute to the learning from demonstration framework when kinesthetic teaching is being used. Here we intend to add visual information to the data on robot kinematics and dynamics data so that the results obtained will be better than those from a single source.

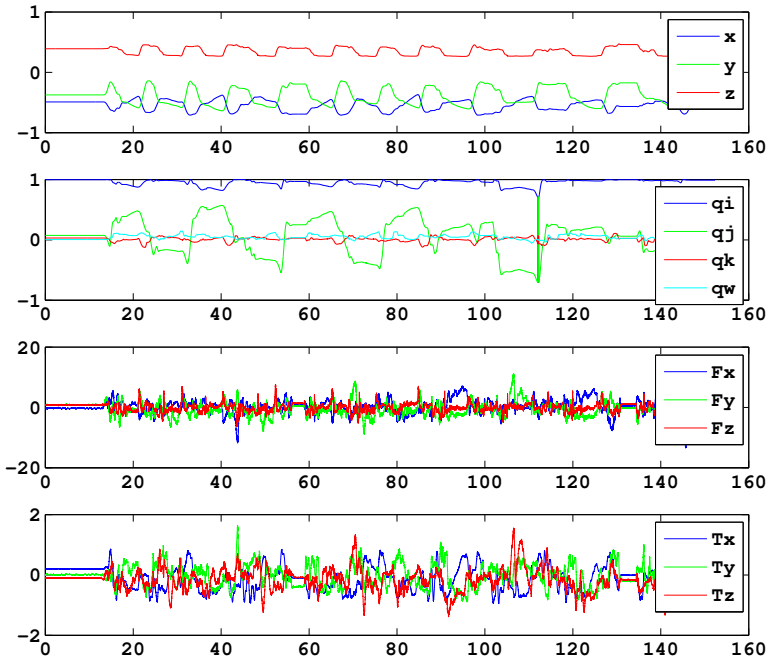


Figure C.21: Robot data extracted from a *rolling* task demonstration. Cartesian (x,y,z) position as well as joint (q_i,q_j,q_k,q_w) states can be captured from robot sensors. Additionally, forces (F_x,F_y,F_z) and torques (T_x,T_y,T_z) can also be captured when interacting with the environment.

A surgical database (Gao et al., 2014) has been recently used to perform experiments in action segmentation and recognition, and its increasing use suggests that a new and more extensive database should be recorded which could be commonly used worldwide. Since there is no database that brings together task demonstrations in learning from demonstration, we proposed to create a learning from demonstration database on which to base our experimentation. The database is the basis of our experimentation and make use of video and kinematic data, but it includes data from other sources, such as MoCap and an extra RGB-D camera which provides an extra point of view on the task. These information sources allow us to perform

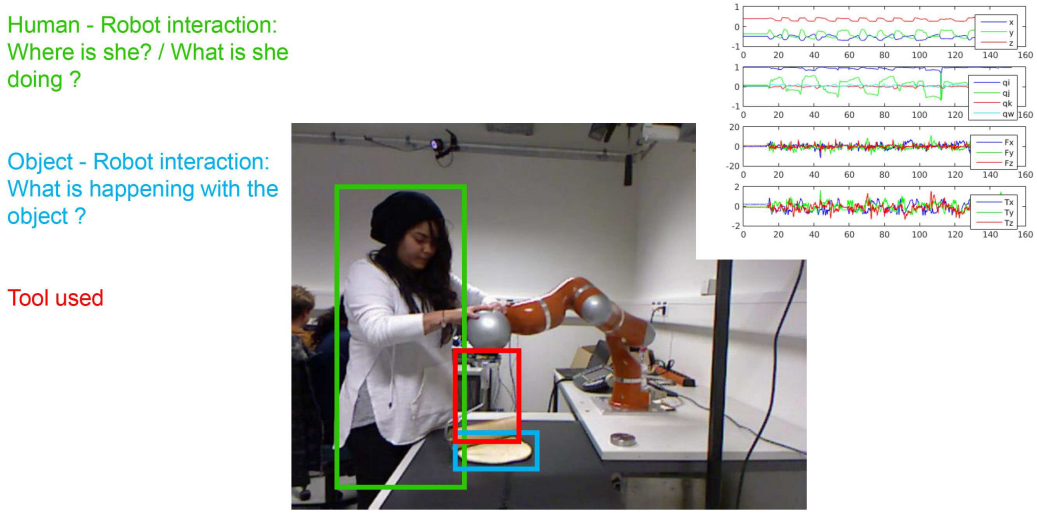


Figure C.22: Information from either the environment and the objects or tools manipulated are incorporated to the system.

some experimentation on the collaborative tasks between human and robot, learning compact models of the interaction. Extracted interaction models can thereafter be used by robots to engage in a similar interaction with a human partner.

C.2.1 Problem Statement

When it comes to decomposing complex tasks from kinesthetic teaching, the automatic segmentation algorithms developed in this LASA require carefully recorded data from the robot's sensors to achieve the desired task decomposition, i.e. the teacher records batches of data that are most useful for motion learning. However, if the goal is for naive users to teach robots in a seamless manner, the procedure must be more autonomous, and be able to identify a set of high-level "teaching" actions being executed by the human during an uninterrupted demonstration, such as:

1. Actions that are relevant for motion learning (i.e. kinesthetic motion guidance), which can be decomposed into:
   (a) Motions involving contact with environment/object.
   (b) Motions in free space.
2. Idle behaviors
3. Actions that are relevant for task goal/metric learning:

(a) Human manipulation of the environment/objects.

(b) Human manipulation/reconfiguration of the robot's end-effector/tool.

Using an action recognition approach the system recognizes every sub-action present in the videos. This is done by fusing different sources of information, such as RGB, Depth videos, Torque and velocity joint values of the robot and MoCap system. This action recognition part requires a new database to be created (actually, recorded data should include the information from all sensors and the data synchronization). The new database includes information from RGB-D cameras from different viewpoints (robot view and environmental view), kinematic and dynamic information from the robot arm, and MoCap system. The most complete action database nowadays includes cameras (RGB-D), microphones, MoCap and accelerometers. The new feature of the recorded database is that it contains information from the robot sensors (torques and velocities), information that can discriminate visually similar actions, such as reaching or rolling. Figure C.23 shows a diagram of the data acquisition system.

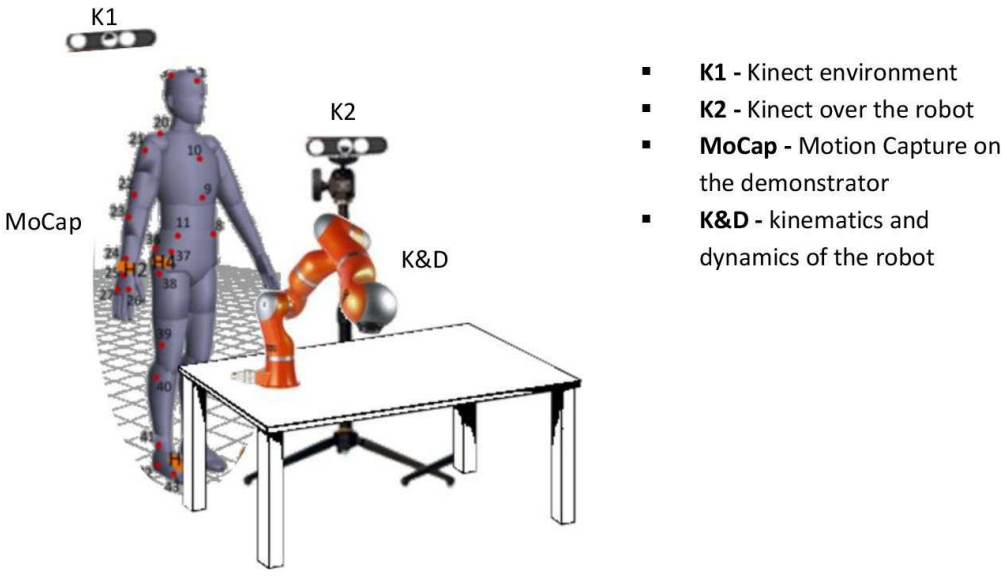


Figure C.23: The data acquisition system.

Taking into account that each of these actions belong to a class, the Action Recognition based on BoW (ARBoW) engine decomposes a complete uninterrupted raw recording from the demonstration of a complex task and automatically selects

the batches of data that are used for segmentation and motion learning. Moreover, the recognition system classifies motions involving contact/no contact (by fusing visual information with robot sensor measurements), used therefore as a prior for the number of states expected in the Beta Process Hidden Markov Model (BP-HMM) algorithm and with the extraction of the control variables as well. See Figure C.24 for schematic details of the approach. Furthermore, once the action recognition engine is in place, we use this new high-level classification of the demonstration to learn an objective function for the goal of the task, by extracting the relevant sequence of actions and states that can help accomplish the goal.

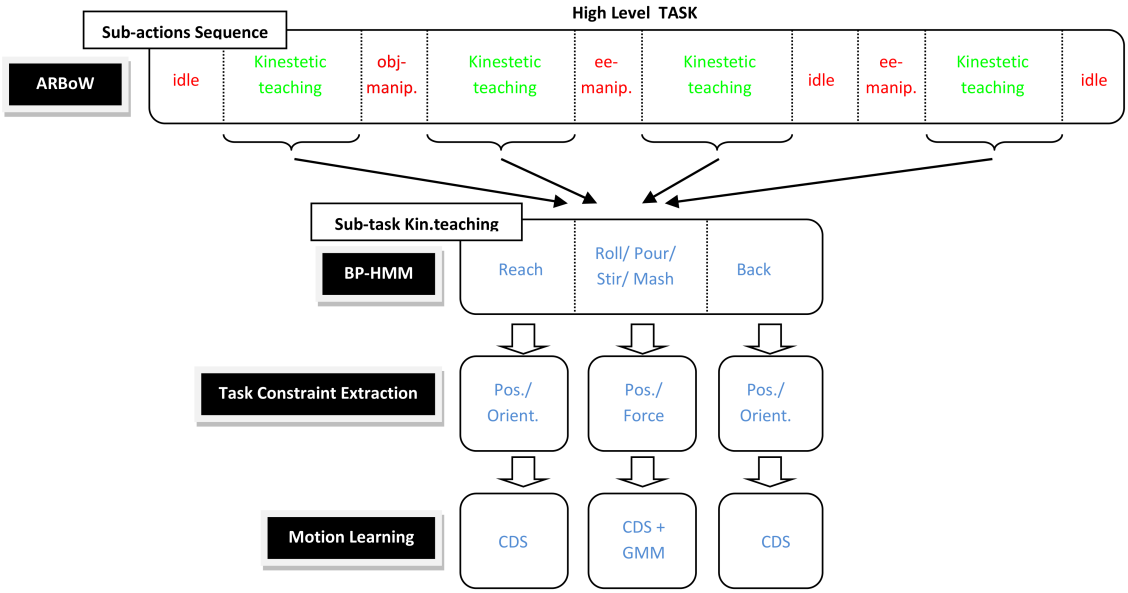


Figure C.24: System overview.

C.2.2 Database

Capturing data to build a database is a non trivial task. In our case we want to record information to teach different high level cooking tasks, such as “rolling”, “pouring”, “stirring” and “mashing”. The experimental setup includes two RGB-D cameras from two different viewpoints, one capturing the environment and the other the robot’s perspective. It also includes data from the robot: namely, position, orientation, force and torque of the end effector. Camera data is recorded at 30

Table C.1: Atomic actions to be segmented from an entire demonstration.

Atomic Action Label	Action description
51	Idle
52	End effector reorientation
53	Objet manipulation
54	Kinestetic teaching

frames per second, while robot data is saved at 500Hz, which means that one-to-one mapping cannot be performed and that an accurate synchronization step is required. All the data is recorded in a ROS bag file, in which every topic recorded keep the information of one sensor with its timestamps.

In order to synchronize the sensor data, we extract the information from each topic saved in the rosbag file. We take the topic with the lowest capture rate and keep its timestamps. We execute a kNN algorithm to match these timestamps with the ones from the other topics. With this step, we synchronize data from one topic to another. After the synchronization, we can extract data from the rosbag file and build the database with RGB videos, Depth data and robot data.

The EPFL Learning from Demonstration Database (EPFL-LfDD) is expected to include data on four elementary cooking tasks performed by 3 subjects. Every subject would perform every task at least 5 times. Therefore, 60 action sequences will be recorded for a total of about 90 minutes recording time. Furthermore, each task will be performed as a sequence of atomic-actions, which will include at least 4 of them per task and not all of them are shared by each task. The specified set of tasks will consist of traditional cooking tasks: (1) Rolling , (2) Pouring , (3) Stirring (4) Mashing and the set of atomic-actions consists of: (1) actions that are important for motion learning (i.e. kinesthetic motion guidance): e.g., motions involving contact with the environment, motions involving contact with the object, motions in free space; (2) actions that are important for task goal learning: e.g., human manipulation of the environment, human manipulation of the objects, human manipulation of the robot's tool; and (3) idle behaviors. This set of atomic actions are listed in Table C.1.

The database will be recorded with two RGB-D cameras from different viewpoints, (1) environment, (2) robot viewpoint. These two viewpoints ensure that there is variability in the parts of the body that are occluded, since the robot viewpoint only sees the arms, hands, end effector, tool and object manipulated. This viewpoint will provide useful information on the transformation of the object which can be used to determine the goal of the task. To include information about human and robot motion, we will record kinematic and dynamics information from the robot and human motion by using the Motion Capture (MoCap) system.

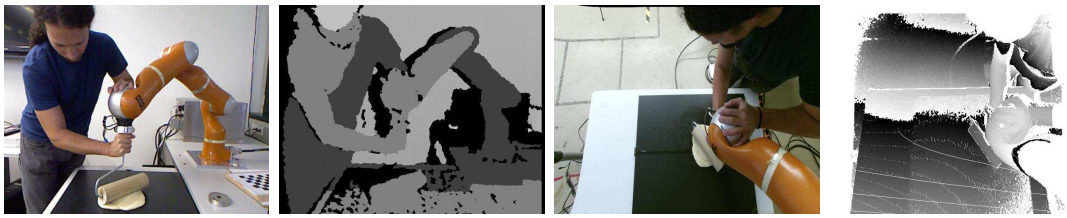


Figure C.25: Example frames taken from the database. From left to right, Kinect RGB, Kinect Depth, Kinec2 RGB, Kinect2 Depth.

Video data is captured by means of two RGB-D cameras. Example frames captured with both cameras can be seen in Figure C.25. The first camera is an environment camera, which captures all the semantic information related to the task performed. This camera is a Kinect from Miscrosoft© with a color camera resolution of 640x480 pixels, which means that the transfer data ratio should be 640x480x30 = 9,216 Mb/s. Since depth camera have less resolution, the ratio for this depth camera is 320x240x30 = 2,304 Mb/s. A second camera is mounted just over the robot arm. This camera will be useful to follow and establish a goal/end of task metric. This camera is a Kinect2 from Microsoft© with a color camera resolution of 1920x1080, which means that the transfer ratio should be at least 1920x1080x30 = 62,208 Mb/s. In this case, the depth camera also has less resolution, so the transfer rate for this depth camera should be 512x424x30 = 6,513 Mb/s. To sum up, the total amount of data transferred by the Kinect camera is 11.52 Mb/s, and by the Kinect2 is 68,721Mb/s. In this case, both USB 2.0 and USB 3.0 (60Mb/s and 640Mb/s respectively) have a good enough data transfer rate to afford all the data. However, considering that a Hard Disk Drive (HDD) has an approximate data transfer rate

of 100Mb/s, this might fall short when there are more processes running on the same HDD, and so it is. This is why we considered using two PCs to perform the acquisition, running a ROS Master / Slave architecture synchronized with the free licensed program Chrony.

Robot kinematic and dynamic data is captured using ROS nodes that capture joint states, including angular position and torques, and the end effector position and forces. This data is collected at 500Hz and synchronized with video frames by means of a kNN algorithm. For each video frame, we set a corresponding window of n robot data rows using the timestamp correspondences. We precompute six statistical measures from every set of windowed raw data, to form a 78 dimensional descriptor as we have 6 force/torque and 7 position measures.The windows are subsets of robot raw data in which every window contains the data with timestamps between two consecutive video frames. Hence, if the robot data is captured at 500Hz, and video data is captured at 30 frames/s, then we have $\tilde{1}9$ robot measures per frame. The robot descriptor building procedure is detailed in FigureC.26.

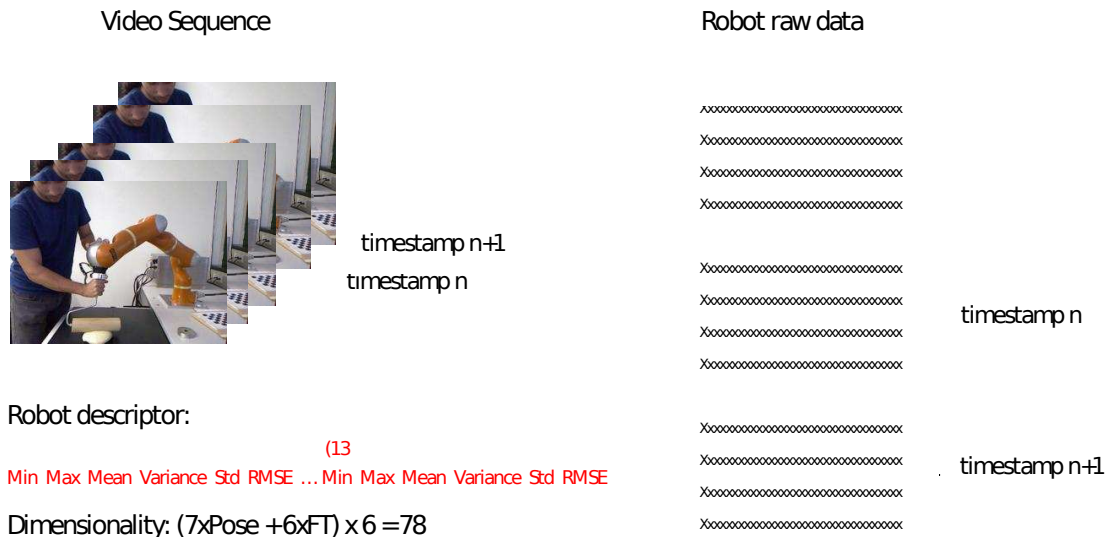


Figure C.26: Robot data descriptor construction.

The EPFL-LfDD is provided with a full manually annotated ground truth for atomic task activity segments. We specified and labeled 5 atomic actions which are used to perform each of the high level tasks. However, not all the atomic actions are

used to perform each task just a subset of them. Each annotation includes the name of the video sequence, the name of the atomic action and a sequence of start-end frames in which the action is performed.

C.2.3 Methodology and Objectives

The ARBoW engine is based on the Bag of Words approach. The method extracts different features that represent a specific kind of information. For example, in RGB videos, the extraction of HOG descriptors for each frame provides spatial information and the use of trajectory descriptors enables time information to be incorporated. For the robot data, we compute a statistical based descriptor from the raw data files. In the end, we have three sets of descriptors, two of which are extracted from video frames, namely HOG and trajectories, and one of which built from robot data. For each set of features we compute a k-means clustering algorithm to find the most relevant k features. Although we know that the higher the codebook word number the better the performance, we are also aware of the curse of dimensionality. For these reason, the number of clusters is set to 100. Our previous experience showed that this could be big enough if redundancy is treated appropriately. Therefore, the codebook generated is used to encode every frame in three different histograms of codeword appearances. Subsequently, they are fused in a multikernel SVM framework. To perform the fusion, we take into account the redundancy introduced when they are combined and to reinforce the important cues, the kernel weights are learned in a gradient descent approach. The overall scheme can be seen in FigureC.27.

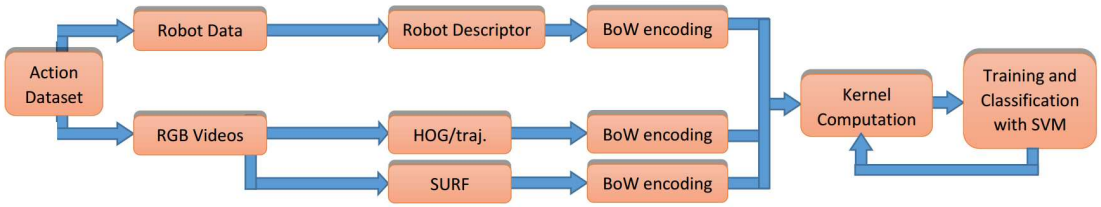


Figure C.27: Overall scheme of the ARBoW engine.

The final goal of applying an action recognition engine is two fold. First, to get the best action recognition engine for extracting the atomic actions of the task and

perform action segmentation based on BP-HMM for the extraction of the task control law. And second, to use this new high-level classification of the demonstration in order to learn an objective function for the goal of the task, by extracting the relevant sequence of actions and states that can lead towards accomplishing the demonstrated goal. In order to achieve these aims we performed different experiments:

1. We compared the results by using data sources separately. First, we used HOG and trajectory descriptors from video sequences, and then robot descriptors built from robot measurements.
2. We compared the results by using a combination of data sources. We combined data from both sensors by using a multiple kernel Support Vector Machine.
3. We ran a test during task execution. We used the video sequence of a whole task (around 1.5 min) and ran the ARBoW engine to get the labels for the action recognized in each frame. We performed the test for two entire sequences separately.
4. We performed a test from a completely new capture. We trained our model using the data recorded in specific conditions. With this experiment we intended to demonstrate that the model learned can be used for new data from other captures.

C.2.4 Preliminary Results

In the first experiment, we used a 10 fold cross validation strategy on the whole bunch of data recorded to analyze the results. We used the traditional one-against-all approach. Other approaches like leave-one-subject-out or leave-one-supertrial-out were not considered at this point because we only had a few subjects and trials.

We first performed a simple recognition task involving single sources of information (i.e. camera captures or robot data). By using spatial (HOG) and temporal (trajectories) information present in the videos, the accuracy is of 89.35%. This contrasts to the accuracy obtained when only robot data is used, which is significantly lower (71,04%). These results are presented in Table C.2. However, with respect to the object manipulation action, there exists full confusion when

Table C.2: Average accuracy for experiments 1-2.

Sensor used	Accuracy (%)	Kernel weights
Kinect2 (HOG + trajectories)	89.35	-
Robot	71.04	-
Kinect2 + Robot	95.13	0.68 - 0.57 - 0.66

using video data only. Thus, there are no label assignments to this action during classification step and this can be a strong issue to face off. Robot data, on the other hand, proved to be able to solve this problem. Hence, it is clear that data fusion is fundamental.

C.2.5 Future Work

- We need more data if better models are to be trained. The two sequences we have been working on so far are quite challenging and the experiments have shown promising results.
- The robot descriptor used for experimentation could be improved by adding joint states (position, velocity and torque).
- We could extend the sequences to other tasks, such as pouring or stirring. We may also consider the possibility of increasing the number of demonstrators and changing the illumination, space and environmental conditions (outdoors is not considered because the kinect performs poorly in these environments, and because the tasks under study are usually performed indoors).
- We could investigate the use of the master-slave configuration to obtain as much data as possible. To prevent data loss, the first step would be to use an Solid State Disk (SSD) instead of an HDD, because its data transfer rate is 10 times higher (1Gb/s compared to 100Mb/s).
- We could use the robot viewpoint to extract the objective function which defines the end of the task. This could be done by capturing high-level features relating the robot and the object parts in an image.
- In order to improve the accuracy of the action recognition engine and make

the algorithm more robust, we could take into account the transitions between actions since the atomic actions are recorded in a sequential order. (This would improve the classifier itself, which is much more interesting than applying a kind of filter at the end of the test)

C.2.6 Testing Tool

A testing tool has been developed to show the results. Although it is in the early stages, the parameters used for the engine and the true / false positives throughout the sequence tested can be seen. The appearance of the framework is shown in FigureC.28.

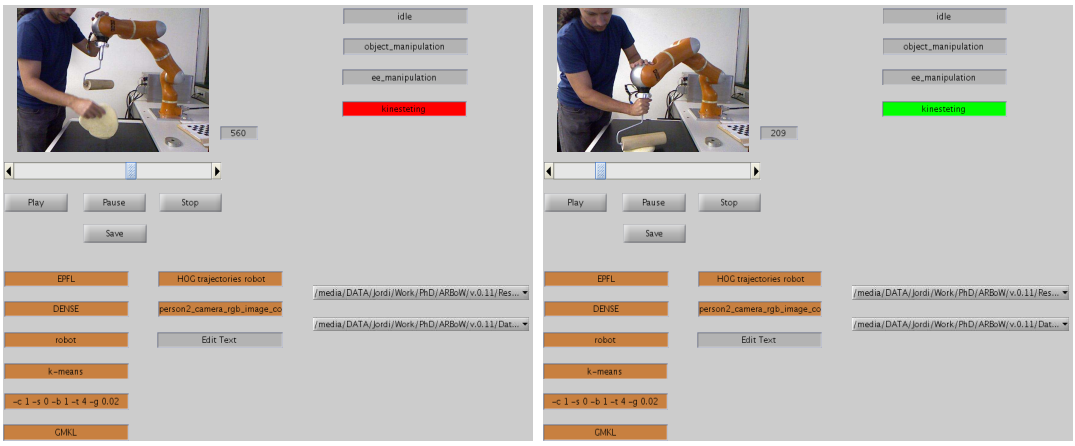


Figure C.28: Testing Tool v.0.1. With this test tool which parameters used by the engine and the label predicted are shown in each instant of the sequence. If there is a good match between the predicted label and the ground truth, the atomic action is highlighted in green. Otherwise, it is highlighted in red (false positive).

A filter can be applied to the predicted labels after the test. Due to the fact that one atomic action must have at least 10 frames (a requisite if the engine is to work properly), we can build a voting filter that takes into account the label with the second highest probability and the frame neighbors. With this filter we do not expect to increase the overall accuracy much more, but we can expect to correct around 1% of the false positives.

Appendix D

Industrial Doctorate

The fact that I have been able to hold down a job while I have been involved in the Industrial PhD programme has been of great value to my professional life because the system has bridged the gap between traditional doctorates and industrial projects. So, I value very positively the experience that I have gained through my involvement with the IRCV laboratory from the URV, where I was able to develop such research skills as searching for information quickly and efficiently, assimilating complex information, analyzing and solving problems, and defending my conclusions. Likewise, my participation in the project VinBot has enabled me to reinforce my professional skills for example, leading teams, working with multidisciplinary and culturally diverse people, managing complex situations, negotiating and reaching consensus, supervising the work of others, and identifying project objectives and managing them and extend them to R+D+i. However, the added difficulty that has led to so much work, and my biggest criticism, is that the objectives of the Industrial Doctorate were not well defined, that the agreement between the parties was not sufficient and many loose ends remained. All of these factors directly or indirectly affected the candidate's final performance.

I should point out that the relationship with the research group from Dr. Domènec Puig's laboratory has enabled a new line of research to be opened up into the recognition of actions and human-machine interaction. Thanks to the industrial doctorate funding other students have been brought into the program and join in

the research effort, which has made it possible to extend the contribution of the group beyond this doctoral thesis. Likewise, my role as research engineer in the FP7 VinBot project required me to be able to work with a wide range of people while commanding respect, having a positive attitude in my dealings with superiors and colleagues, and assuming my responsibility for all the tasks entrusted to me.

Finally, I would like to emphasize my personal and professional growth during these three years, which is the result of the variety of experiences I have been subject to at both the URV and Ateknea. During this time, I have developed my leadership skills, and been involved to a greater or lesser extent in coordination and management of R+D+i projects, transfer of research results, development of new businesses (entrepreneurship, business management, financing sources) and patents, intellectual and industrial property. I am aware that this variety has provided me with a wealth of experience that is of great professional and personal value.

Goals attained

- FP7 VinBot: We have built a robot platform that has acquired about 650Gb of data from the growth stages of vines. The platform was built in three stages: the first had the basic elements that allowed it to function; the second included ROS middleware which enabled us to synchronize the sensors and parts much more easily; and the third excluded the redundant hardware elements and included post-processing in the cloud (Amazon EC2). With this platform we saved visual data from RGB and NIR cameras, and also 3D data using a rotating laser depth device. The platform was tested in several demonstrations at fairs and meetings of the consortium and for a year and a half while the data was being acquired.
- FP7 RoboHow: I spent a research period at an internationally recognized research center with the main objective of working in a multicultural and multidisciplinary environment, learning techniques developed by one of the world's top laboratories. We obtained data from everyday kitchen tasks by

recording information of the robot kinematics and dynamics thanks to its force and motion sensors, as well as recording visual information with both first- and second-generation RGB-D cameras. We improved the recognition algorithm in order to aggregate data from the robot and the action recognition engine was applied to the data for the segmentation of the task and the extraction of sub-action labels that represent the entire task.

- Publications: scientific papers have been published in accordance with the university's requirements and the stipulations of both the Consortium and Grant agreements of the VinBot project. The intellectual property rights that apply to the results obtained in the VinBot project allow for dissemination and publication only if this is compatible with the legitimate interests of the owners of exploitation and commercialization of these results.
- Cross-training activities. I engaged in a total of 109 hours of cross-training activities: 30 hours of general training given by the Government on various topics; 19 hours on leadership and intellectual property; and finally 60 hours on entrepreneurship and leadership.

Understanding My Transferable Skills

When considering a career outside academia, I need to think in terms other than academic labels and academic signifiers of success, such as the number of publications or talks I have been invited to give. Instead, I need to focus on the skills I used to earn my degree and that are also necessary for a particular job; these skills are my transferable skills and are not restricted to just my research or teaching experience.

There are many approaches to understanding what transferable skills and career interests I have. One approach is to take an assessment such as SkillScan, Myers Briggs Type Indicator, or StrengthsQuest, which can help me identify my skills and potential careers using those skills. Another approach is to analyze my previous experience in an attempt to identify what skills I used or acquired. Table D.1 is a small sample of transferable skills that I have used during my graduate experience.

Table D.1: Generic skills as a Industrial PhD. Table adapted from the Cornell Career Services listing of a PhD's transferable skills.

		Needs Work	Attain	Enjoy
Research and Analytic Skills	Locate and assimilate new information rapidly		X	
	Understand complex information and synthesize it	X		
	Reach independent conclusions and defend them	X		
	Analyze and solve problems			X
Communication Skills	Write clearly at different levels, from abstracts to book-length manuscripts		X	
	Edit and proofread		X	
	Writing and conversing in your second/third language			X
	Speaking in public		X	
	Convey complex information to non-expert audiences			X
Interpersonal Skills	Leadership skills (lab or office)		X	
	Managing individuals		X	
	Working with international colleagues		X	
	Diplomacy and tact (a survival skill in all environments)		X	
	Ability to accept criticism		X	
	Ability to cope with and manage different personalities		X	
	Ability to navigate complex environments		X	
	Persuasion skills (e.g., grant proposals, negotiation within your department)		X	
	Consensus-building skills (e.g., with your department/committee)		X	
	Ability to handle complaints		X	
Organization and Management	Manage your research data and dissertation			X
	Event organization and planning (conferences, programs, panels)			X
Project Management	Identifying goals and objectives, constraints, timeframes, methodology and stakeholders for a specific project		X	
	Organizing, motivating and controlling resources, procedures and protocols			X
Supervision Skills	Evaluated others' performance		X	
	Monitored or oversaw the work of others in a lab, field, institute or office		X	
Personal skills	Intellectual strength and courage			X
	Perform under pressure		X	
	Meet deadlines		X	
	Focus, tenacity, stamina, and discipline		X	
	Self-reliance, autonomy			X
	See a task through to completion			X
Entrepreneurial Skills	Think creatively			X
	Acquire funding (e.g., write grant proposals)	X		
	Manage a budget	X		

References

Abbeel, P., Coates, A., Quigley, M., and Ng, A. Y. (2007). An application of reinforcement learning to aerobatic helicopter flight. *Advances in neural information processing systems*, 19:1.

Abbeel, P. and Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM.

Agarwal, S., Saradhi, V. V., and Karnick, H. (2008). Kernel-based online machine learning and support vector reduction. *Neurocomputing*, 71(79):1230 – 1237. Progress in Modeling, Theory, and Application of Computational Intelligenc15th European Symposium on Artificial Neural Networks 200715th European Symposium on Artificial Neural Networks 2007.

Aggarwal, J. K. and Ryoo, M. S. (2011). Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16.

Aleotti, J. and Caselli, S. (2006). Robust trajectory learning and approximation for robot programming by demonstration. *Robotics and Autonomous Systems*, 54(5):409–413.

Argall, B., Browning, B., and Veloso, M. (2007). Learning by demonstration with critique from a human teacher. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 57–64. ACM.

Argall, B. D., Browning, B., and Veloso, M. M. (2011a). Teacher feedback to scaffold and refine demonstrated motion primitives on a mobile robot. *Robotics and Autonomous Systems*, 59(3):243–255.

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469 – 483.

Argall, B. D., Sauser, E. L., and Billard, A. G. (2011b). *Tactile guidance for policy adaptation*, volume 2. Now Publishers Inc.

Arnó, J., Vallès, J. M., Llorens, J., Sanz, R., Masip, J., Palacín, J., Rosell-Polo, J. R., et al. (2013). Leaf area index estimation in vineyards using a ground-based lidar scanner. *Precision agriculture*, 14(3):290–306.

Bilinski, P. and Corvee, E. (2013). Relative Dense Tracklets for Human Action Recognition. *10th IEEE International Conference on Automatic Face and Gesture Recognition*.

Billard, A. and Matarić, M. J. (2001). Learning human arm movements by imitation:: Evaluation of a biologically inspired connectionist architecture. *Robotics and Autonomous Systems*, 37(2):145–160.

Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. (2005). Actions as space-time shapes. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1395–1402. IEEE.

Bobick, A. F. and Davis, J. W. (2001). The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(3):257–267.

Bobick, A. F. and Wilson, A. D. (1997). A state-based approach to the representation and recognition of gesture. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(12):1325–1337.

Boley, D. and Cao, D. (2004). Training support vector machines using adaptive clustering. In *SDM*, pages 126–137. SIAM.

Bordes, A., Ertekin, S., Weston, J., and Bottou, L. (2005). Fast kernel classifiers with online and active learning. *J. Mach. Learn. Res.*, 6:1579–1619.

Boukharouba, K., Bako, L., and Lecoeuche, S. (2009). Incremental and decremental multi-category classification by support vector machines. In *Machine Learning and Applications, 2009. ICMLA'09. International Conference on*, pages 294–300. IEEE.

Breazeal, C., Berlin, M., Brooks, A., Gray, J., and Thomaz, A. L. (2006). Using perspective taking to learn from ambiguous demonstrations. *Robotics and autonomous systems*, 54(5):385–393.

Breazeal, C. and Scassellati, B. (2002). Robots that imitate humans. *Trends in Cognitive Sciences*, 6(11):481 – 487.

Bucak, S., Jin, R., and Jain, A. (2014). Multiple kernel learning for visual object recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(7):1354–1369.

Bucak, S., Jin, R., and Jain, A. K. (2010). Multi-label multiple kernel learning by stochastic approximation: Application to visual object recognition. In *Advances in Neural Information Processing Systems*, pages 325–333.

Calinon, S. and Billard, A. (2007). Incremental learning of gestures by imitation in a humanoid robot. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 255–262. ACM.

Calinon, S., D'halluin, F., Sauser, E. L., Caldwell, D. G., and Billard, A. G. (2010a). Learning and reproduction of gestures by imitation. *Robotics & Automation Magazine, IEEE*, 17(2):44–54.

Calinon, S., Sauser, E. L., Billard, A. G., and Caldwell, D. G. (2010b). Evaluation of a probabilistic approach to learn and reproduce gestures by imitation. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2671–2676. IEEE.

Campbell, L. W. and Bobick, A. E. (1995). Recognition of human body motion using phase space constraints. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 624–630. IEEE.

Cangelosi, A., Metta, G., Sagerer, G., Nolfi, S., Nehaniv, C., Fischer, K., Tani, J., Belpaeme, T., Sandini, G., Nori, F., et al. (2010). Integration of action and language knowledge: A roadmap for developmental robotics. *Autonomous Mental Development, IEEE Transactions on*, 2(3):167–195.

Cauwenberghs, G. and Poggio, T. (2001). Incremental and decremental support vector machine learning. *Adv. Neural Information Processing Systems (NIPS*2000)*, 13.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Chernova, S. and Veloso, M. (2007). Confidence-based policy learning from demonstration using gaussian mixture models. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 233. ACM.

Chernova, S. and Veloso, M. (2008a). Learning equivalent action choices from demonstration. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1216–1221. IEEE.

Chernova, S. and Veloso, M. (2008b). Multi-thresholded approach to demonstration selection for interactive robot learning. In *Human-Robot Interaction (HRI), 2008 3rd ACM/IEEE International Conference on*, pages 225–232. IEEE.

Chernova, S. and Veloso, M. (2009). Interactive policy learning through confidence-based autonomy. *J. Artif. Int. Res.*, 34(1):1–25.

Chersi, F. (2012). Learning through imitation: a biological approach to robotics. *Autonomous Mental Development, IEEE Transactions on*, 4(3):204–214.

Choi, J. and Kim, K. (2012). Nonparametric bayesian inverse reinforcement learning for multiple reward functions. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 314–322.

Chomat, O. and Crowley, J. L. (1999). Probabilistic recognition of activity using local appearance. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE.

Coates, A., Abbeel, P., and Ng, A. Y. (2008). Learning for control from multiple demonstrations. In *Proceedings of the 25th international conference on Machine learning*, pages 144–151. ACM.

Comaniciu, D. and Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619.

Cour, T., Jordan, C., Miltsakaki, E., and Taskar, B. (2008). Movie/script: Alignment and parsing of video and text transcription. In *Computer Vision–ECCV 2008*, pages 158–171. Springer.

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

Cupillard, F., Brémond, F., and Thonnat, M. (2002). Group behavior recognition with multiple cameras. In *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*, pages 177–183. IEEE.

Dai, P., Di, H., Dong, L., Tao, L., and Xu, G. (2008). Group interaction analysis in dynamic context. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 38(1):275–282.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 886–893, Washington, DC, USA. IEEE Computer Society.

Dalal, N., Triggs, B., and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part II*, ECCV'06, pages 428–441, Berlin, Heidelberg. Springer-Verlag.

Damen, D. and Hogg, D. (2009). Recognizing linked events: Searching the space of feasible explanations. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 927–934. IEEE.

Darrell, T. and Pentland, A. (1993). Space-time gestures. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on*, pages 335–340. IEEE.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Dhillon, I. S., Guan, Y., and Kulis, B. (2007). Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(11):1944–1957.

Diehl, C. P. and Cauwenberghs, G. (2003). Svm incremental learning, adaptation and optimization. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 4, pages 2685–2690. IEEE.

Dollar, P., Rabaud, V., Cottrell, G., and Belongie, S. (2005). 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. IEEE Computer Society.

Duan, H., Shao, X., Hou, W., He, G., and Zeng, Q. (2009). An incremental learning algorithm for lagrangian support vector machines. *Pattern Recognition Letters*, 30(15):1384 – 1391.

Duchenne, O., Laptev, I., Sivic, J., Bach, F., and Ponce, J. (2009). Automatic annotation of human actions in video. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1491–1498. IEEE.

Efros, A. A., Berg, A. C., Mori, G., and Malik, J. (2003). Recognizing action at a distance. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 726–733. IEEE.

Farhadi, A. and Tabrizi, M. K. (2008). Learning to recognize activities from the wrong view point. In *Computer Vision–ECCV 2008*, pages 154–166. Springer.

Farhadi, A., Tabrizi, M. K., Endres, I., and Forsyth, D. (2009). A latent model of discriminative aspect. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 948–955. IEEE.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.

Frieß, T.-T., Cristianini, N., and Campbell, C. (1998). The kernel-adatron algorithm: A fast and simple learning procedure for support vector machines. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 188–196, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Gao, Y., Vedula, S. S., Reiley, C. E., Ahmidi, N., Varadarajan, B., Lin, H. C., Tao, L., Zappella, L., Béjar, B., Yuh, D. D., et al. (2014). Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling. *Modeling and Monitoring of Computer Assisted Interventions*, pages 1–10.

Gavrila, D., Davis, L., et al. (1995). Towards 3-d model-based tracking and recognition of human movement: a multi-view approach. In *International workshop on automatic face-and gesture-recognition*, pages 272–277. Citeseer.

Gehler, P. and Nowozin, S. (2009). Let the kernel figure it out; principled learning of pre-processing for kernel classifiers. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 2836–2843. IEEE.

Ghanem, N., DeMenthon, D., Doermann, D., and Davis, L. (2004). Representation and recognition of events in surveillance video using petri nets. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*, pages 112–112. IEEE.

Gong, S. and Xiang, T. (2003). Recognition of group activities using dynamic probabilistic networks. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 742–749. IEEE.

Grollman, D. and Billard, A. (2011). Donut as i do: Learning from failed demonstrations. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3804–3809.

Grollman, D. and Jenkins, O. (2010). Incremental learning of subtasks from unsegmented demonstration. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 261–266.

Grollman, D. H. and Jenkins, O. C. (2007). Dogged learning for robots. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2483–2488. IEEE.

Grollman, D. H. and Jenkins, O. C. (2008). Sparse incremental learning for interactive robot control policy estimation. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3315–3320. IEEE.

Guenter, F. and Billard, A. G. (2007). Using reinforcement learning to adapt an imitation task. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1022–1027. IEEE.

Gupta, A. and Davis, L. S. (2007). Objects in action: An approach for combining action understanding and object perception. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE.

Gupta, A., Srinivasan, P., Shi, J., and Davis, L. S. (2009). Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2012–2019. IEEE.

Gupta, S. and Mooney, R. J. (2009). Using closed captions to train activity recognizers that improve video retrieval. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 30–37. IEEE.

Hai, Y., He, W., and Fan, L. (2010). An incremental learning algorithm for svm based on voting principle. In *Advanced Information Management and Service (IMS), 2010 6th International Conference on*, pages 420–423.

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

Hoai, M., Lan, Z.-Z., and De la Torre, F. (2011). Joint segmentation and classification of human actions in video. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3265–3272. IEEE.

Hu, Y., Cao, L., Lv, F., Yan, S., Gong, Y., and Huang, T. S. (2009). Action detection in complex scenes with spatial and temporal ambiguities. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 128–135. IEEE.

Ijspeert, A. J., Nakanishi, J., and Schaal, S. (2001). Trajectory formation for imitation with nonlinear dynamical systems. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 2, pages 752–757. IEEE.

Ijspeert, A. J., Nakanishi, J., and Schaal, S. (2002a). Learning rhythmic movements by demonstration using nonlinear oscillators. In *Proceedings of the ieee/rsj int. conference on intelligent robots and systems (iros2002)*, pages 958–963.

Ijspeert, A. J., Nakanishi, J., and Schaal, S. (2002b). Movement imitation with nonlinear dynamical systems in humanoid robots. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 2, pages 1398–1403.

Ikizler-Cinbis, N., Cinbis, R. G., and Sclaroff, S. (2009). Learning actions from the web. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 995–1002. IEEE.

Ikizler-Cinbis, N. and Sclaroff, S. (2010). Object, scene and actions: Combining multiple features for human action recognition. In *Proceedings of the 11th European Conference on Computer Vision: Part I*, ECCV'10, pages 494–507, Berlin, Heidelberg. Springer-Verlag.

Intille, S. S. and Bobick, A. F. (1999). A framework for recognizing multi-agent action from visual evidence. *AAAI/IAAI*, 99:518–525.

Ivanov, Y. A. and Bobick, A. F. (2000). Recognition of visual activities and interactions by stochastic parsing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):852–872.

Jiang, H., Drew, M. S., and Li, Z.-N. (2006). Successive convex matching for action detection. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1646–1653. IEEE.

Jiang, Y., Dai, Q., Xue, X., Liu, W., and Ngo, C. (2012). Trajectory-based modeling of human actions with motion reference points. In *European Conference on Computer Vision (ECCV)*.

Joo, S.-W. and Chellappa, R. (2006). Attribute grammar-based event recognition

and anomaly detection. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 107–107. IEEE.

Junejo, I. N., Dexter, E., Laptev, I., and Pérez, P. (2008). *Cross-view action recognition from temporal self-similarities*. Springer.

Karaoguz, C., Rodemann, T., Wrede, B., and Goerick, C. (2013). Learning information acquisition for multitasking scenarios in dynamic environments. *Autonomous Mental Development, IEEE Transactions on*, 5(1):46–61.

Ke, Y., Sukthankar, R., and Hebert, M. (2007). Spatio-temporal shape and flow correlation for action recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE.

Khan, S. M. and Shah, M. (2005). Detecting group activities using rigidity of formation. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 403–406. ACM.

Khansari-Zadeh, S. M. and Billard, A. (2010). Bm: An iterative algorithm to learn stable non-linear dynamical systems with gaussian mixture models. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2381–2388. IEEE.

Khansari-Zadeh, S. M. and Billard, A. (2011). Learning stable nonlinear dynamical systems with gaussian mixture models. *Robotics, IEEE Transactions on*, 27(5):943–957.

Kivinen, J., Smola, A., and Williamson, R. (2004). Online learning with kernels. *Signal Processing, IEEE Transactions on*, 52(8):2165–2176.

Kläser, A., Marszałek, M., and Schmid, C. (2008). A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conference*, pages 995–1004.

Kober, J. and Peters, J. (2009). Learning motor primitives for robotics. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2112–2118. IEEE.

Koppula, H. S., Gupta, R., and Saxena, A. (2013). Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T. (2011). HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Lanckriet, G. R. G., Cristianini, N., Bartlett, P., El Ghaoui, L., and Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72.

Laptev, I. (2005). On space-time interest points. *Int. J. Comput. Vision*, 64(2-3):107–123.

Laptev, I., Caputo, B., Schüldt, C., and Lindeberg, T. (2007). Local velocity-adapted motion events for spatio-temporal recognition. *Computer Vision and Image Understanding*, 108(3):207–229.

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

Li, B., Chi, M., Fan, J., and Xue, X. (2007). Support cluster machine. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 505–512, New York, NY, USA. ACM.

Liang, Z. and Li, Y. (2009). Incremental support vector machine learning in the primal and applications. *Neurocomputing*, 72(1012):2249 – 2258. Lattice Computing and Natural Computing (JCIS 2007) / Neural Networks in Intelligent Systems Designn (ISDA 2007).

Liu, K. and Yang, J. (2010). Online recognition of people recurrences with bag-of-features representation and automatic new-class labeling. *Optical Engineering*, 49(1):017203–017203–10.

Lockerd, A. and Breazeal, C. (2004). Tutelage and socially guided robot learning. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 4, pages 3475–3480. IEEE.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee.

Lublinerman, R., Özay, N., Zarpalas, D., and Camps, O. (2006). Activity recognition from silhouettes using linear systems and model (in) validation techniques. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 347–350. IEEE.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lv, F., Kang, J., Nevatia, R., Cohen, I., and Medioni, G. (2004). Automatic tracking and labeling of human activities in a video sequence. In *Proceedings of the 6th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS04)*.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif. University of California Press.

Marszalek, M., Laptev, I., and Schmid, C. (2009). Actions in context. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2929–2936. IEEE.

Merrick, K. E. (2012). Intrinsic motivation and introspection in reinforcement learning. *Autonomous Mental Development, IEEE Transactions on*, 4(4):315–329.

Minnen, D., Essa, I., and Starner, T. (2003). Expectation grammars: Leveraging high-level expectations for activity recognition. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–626. IEEE.

Moore, D. and Essa, I. (2002). Recognizing multitasked activities from video using stochastic context-free grammar. In *AAAI/IAAI*, pages 770–776.

Moore, D. J., Essa, I. A., and Hayes III, M. H. (1999). Exploiting human actions and object context for recognition tasks. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 80–86. IEEE.

Mülling, K., Kober, J., Kroemer, O., and Peters, J. (2013). Learning to select and generalize striking movements in robot table tennis. *International Journal of Robotics Research*, 32(3):263–279.

Natarajan, P. and Nevatia, R. (2007). Coupled hidden semi markov models for activity recognition. In *Motion and Video Computing, 2007. WMVC'07. IEEE Workshop on*, pages 10–10. IEEE.

Nehaniv, C. and Dautenhahn, K. (2001). Like me? - measures of correspondence and imitation. *Cybernetics and Systems*, 32(1-2):11–51. Original article can be found at: http://www.informaworld.com/smpp/title content=t713722751 Copyright Informa / Taylor and Francis Group. DOI: 10.1080/019697201300001803 [Full text of this article is not available in the UHRA].

Nehmzow, U., Akanyeti, O., Weinrich, C., Kyriacou, T., and Billings, S. A. (2007). Robot programming by demonstration through system identification. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 801–806. IEEE.

Nevatia, R., Hobbs, J., and Bolles, B. (2004). An ontology for video event representation. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*, pages 119–119. IEEE.

Nevatia, R., Zhao, T., and Hongeng, S. (2003). Hierarchical language-based representation of events in video streams. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on*, volume 4, pages 39–39. IEEE.

Nguyen, N. T., Phung, D. Q., Venkatesh, S., and Bui, H. (2005). Learning and detecting activities from movement trajectories using the hierarchical hidden markov model. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 955–960. IEEE.

Nicolescu, M., Chadwicke Jenkins, O., Olenderski, A., and Fritzinger, E. (2008). Learning behavior fusion from demonstration. *Interaction Studies*, 9(2):319–352.

Nicolescu, M. N. and Mataric, M. J. (2003). Natural methods for robot task learning: Instructive demonstrations, generalization and practice. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 241–248. ACM.

Niebles, J. C., Wang, H., and Fei-Fei, L. (2008). Unsupervised learning of human action categories using spatial-temporal words. *International journal of computer vision*, 79(3):299–318.

Ogino, M., Toichi, H., Yoshikawa, Y., and Asada, M. (2006). Interaction rule learning with a human partner based on an imitation faculty with a simple visuo-motor mapping. *Robotics and Autonomous Systems*, 54(5):414–418.

Oliver, N., Horvitz, E., and Garg, A. (2002). Layered representations for human activity recognition. In *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pages 3–8. IEEE.

Oliver, N. M., Rosario, B., and Pentland, A. P. (2000). A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):831–843.

Park, S. and Aggarwal, J. K. (2004). A hierarchical bayesian network for event recognition of human actions and interactions. *Multimedia systems*, 10(2):164–179.

Pastor, P., Hoffmann, H., Asfour, T., and Schaal, S. (2009). Learning and generalization of motor skills by learning from demonstration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 763–768. IEEE.

Peursum, P., West, G., and Venkatesh, S. (2005). Combining image regions and human activity for indirect object recognition in indoor wide-angle views. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 82–89. IEEE.

Pieropan, A., Salvi, G., Pauwels, K., and Kjellstrom, H. (2014). Audio-visual classification and detection of human manipulation actions. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3045–3052.

Pinhanez, C. S. and Bobick, A. F. (1998). Human action detection using pnf propagation of temporal constraints. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 898–904. IEEE.

Pook, P. K. and Ballard, D. H. (1993). Recognizing teleoperated manipulations. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 578–585 vol.2.

Poppe, R. (2010). A survey on vision-based human action recognition. *Image Vision Comput.*, 28(6):976–990.

Rakotomamonjy, A., Bach, F. R., Canu, S., and Grandvalet, Y. (2008). Simplemkl. *Journal of Machine Learning Research*.

Ralaivola, L. and d'Alche Buc, F. (2001). Incremental support vector machine learning: A local approach. In Dorffner, G., Bischof, H., and Hornik, K.,

editors, *Artificial Neural Networks ICANN 2001*, volume 2130 of *Lecture Notes in Computer Science*, pages 322–330. Springer Berlin Heidelberg.

Rao, C. and Shah, M. (2001). View-invariance in action recognition. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–316. IEEE.

Ratliff, N., Ziebart, B., Peterson, K., Bagnell, J. A., Ratliff, N., Ziebart, B., Peterson, K., Bagnell, J. A., Hebert, M., Dey, A. K., Srinivasa, S., Ratliff, N., Ziebart, B., Peterson, K., Bagnell, J. A., Hebert, M., Dey, A. K., and Srinivasa, S. (2009). Inverse optimal heuristic control for imitation learning. In *In Proc. of AISTATS*, pages 424–431.

Ratliff, N. D., Bagnell, J. A., and Zinkevich, M. A. (2006). Maximum margin planning. In *Proceedings of the 23rd international conference on Machine learning*, pages 729–736. ACM.

Reddy, K. K. and Shah, M. (2013). Recognizing 50 human action categories of web videos. *Mach. Vision Appl.*, 24(5):971–981.

Rodriguez, M. D., Ahmed, J., and Shah, M. (2008). Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

Rofouei, M., Moazeni, M., and Sarrafzadeh, M. (2008). Fast gpu-based space-time correlation for activity recognition in video sequences. In *Embedded Systems for Real-Time Multimedia, 2008. ESTImedia 2008. IEEE/ACM/IFIP Workshop on*, pages 33–38. IEEE.

Rüping, S. (2001). Incremental learning with support vector machines. In *icdm*, page 641. IEEE.

Rusu, R. B. (2009). *Semantic 3D Object Maps for Everyday Manipulation in Human*

Living Environments. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany.

Rybski, P. E. and Voyles, R. M. (1999). Interactive task training of a mobile robot through human gesture recognition. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 1, pages 664–669. IEEE.

Rybski, P. E., Yoon, K., Stolarz, J., and Veloso, M. M. (2007). Interactive robot task training through dialog and demonstration. In *Human-Robot Interaction (HRI), 2007 2nd ACM/IEEE International Conference on*, pages 49–56. IEEE.

Ryoo, M. and Aggarwal, J. (2008). Recognition of high-level group activities based on activities of individual members. In *Motion and video Computing, 2008. WMVC 2008. IEEE Workshop on*, pages 1–8. IEEE.

Ryoo, M. S. and Aggarwal, J. (2007). Hierarchical recognition of human activities interacting with objects. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE.

Ryoo, M. S. and Aggarwal, J. K. (2009a). Semantic representation and recognition of continued and recursive human activities. *International journal of computer vision*, 82(1):1–24.

Ryoo, M. S. and Aggarwal, J. K. (2009b). 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.

Saunders, J., Nehaniv, C. L., and Dautenhahn, K. (2006). Teaching robots by moulding behavior and scaffolding the environment. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 118–125. ACM.

Schuldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: A local svm approach. In *Proceedings of the Pattern Recognition, 17th International*

Conference on (ICPR'04) Volume 3 - Volume 03, ICPR '04, pages 32–36, Washington, DC, USA. IEEE Computer Society.

Scovanner, P., Ali, S., and Shah, M. (2007). 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. ACM.

Shechtman, E. and Irani, M. (2005). Space-time behavior based correlation. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 405–412. IEEE.

Sheikh, Y., Sheikh, M., and Shah, M. (2005). Exploring the space of a human action. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 144–149. IEEE.

Shi, Y., Huang, Y., Minnen, D., Bobick, A., and Essa, I. (2004). Propagation networks for recognition of partially ordered sequential action. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–862. IEEE.

Siskind, J. M. (2001). Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *J. Artif. Intell. Res.(JAIR)*, 15:31–90.

Smart, W. D. (2002). *Making reinforcement learning work on real robots*. PhD thesis, Brown University.

Snoek, C. G. M., Worring, M., and Smeulders, A. W. M. (2005). Early versus late fusion in semantic video analysis. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, pages 399–402, New York, NY, USA. ACM.

Solmaz, B., Modiri, S. A., and Shah, M. (2012). Classifying web videos using a global video descriptor. *Machine Vision and Applications*.

Starner, T. and Pentland, A. (1997). Real-time american sign language recognition from video using hidden markov models. In *Motion-Based Recognition*, pages 227–243. Springer.

Steil, J. J., Röthling, F., Haschke, R., and Ritter, H. (2004). Situated robot learning for multi-modal instruction and imitation of grasping. *Robotics and Autonomous Systems*, 47(2):129–141.

Suma, E. A., Sinclair, C. W., Babbs, J., and Souvenir, R. (2008). A sketch-based approach for detecting common human actions. In *Advances in Visual Computing*, pages 418–427. Springer.

Syed, N. A., Huan, S., Kah, L., and Sung, K. (1999). Incremental learning with support vector machines. *International Joint Conference on Artificial Intelligence (IJCAI)*.

Tardaguila, J., Diago, M., Millan, B., Blasco, J., Cubero, S., and Aleixos, N. (2012). Applications of computer vision techniques in viticulture to assess canopy features, cluster morphology and berry size. In *I International Workshop on Vineyard Mechanization and Grape and Wine Quality 978*, pages 77–84.

Teixeira, L. F. and Corte-Real, L. (2009). Video object matching across multiple independent views using local descriptors and adaptive learning. *Pattern Recognition Letters*, 30(2):157 – 167. Video-based Object and Event Analysis.

Tsai, J.-S., Hsu, Y.-P., Liu, C., and Fu, L.-C. (2013). An efficient part-based approach to action recognition from rgb-d video with bow-pyramid representation. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2234–2239.

Ude, A., Atkeson, C. G., and Riley, M. (2004). Programming full-body movements for humanoid robots by observation. *Robotics and autonomous systems*, 47(2):93–108.

Vapnik, V. (2000). *The nature of Statistical Learning Theory*. Springer Verlag.

Vaswani, N., Chowdhury, A. R., and Chellappa, R. (2003). Activity recognition using the dynamics of the configuration of interacting objects. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–633. IEEE.

Vedaldi, A., Gulshan, V., Varma, M., and Zisserman, A. (2009). Multiple kernels for object detection. *In Proceedings of the International Conference on Computer Vision, 2009.*

Veeraraghavan, A., Chellappa, R., and Roy-Chowdhury, A. K. (2006). The function space of an activity. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 959–968. IEEE.

Vijayakumar, S. and Schaal, S. (2000). Locally weighted projection regression: An o (n) algorithm for incremental real time learning in high dimensional space. In *International conference on machine learning, proceedings of the sixteenth conference.*

Vu, V.-T., Bremond, F., and Thonnat, M. (2003). Automatic video interpretation: A novel algorithm for temporal scenario recognition. In *IJCAI*, volume 3, pages 1295–1300.

Wang, H., Kläser, A., Schmid, C., and Liu, C. (2013). Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision.*

Wang, H., Kläser, A., Schmid, C., and Liu, C.-L. (2011). Action Recognition by Dense Trajectories. In *IEEE Conf. on Computer Vision & Pattern Recognition*, pages 3169–3176, Colorado Springs, United States.

Wang, H. and Schmid, C. (2013). Action Recognition with Improved Trajectories. In *ICCV 2013 - IEEE International Conference on Computer Vision*, pages 3551–3558, Sydney, Australie. IEEE.

Willems, G., Tuytelaars, T., and Gool, L. (2008). An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proceedings of the 10th European Conf. on Computer Vision: Part II*, ECCV '08, pages 650–663, Berlin, Heidelberg. Springer-Verlag.

Williams, C. K. and Rasmussen, C. E. (2006). Gaussian processes for machine learning. *the MIT Press*, 2(3):4.

Wong, S.-F., Kim, T.-K., and Cipolla, R. (2007). Learning motion categories using both semantic and structural information. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6. IEEE.

Wu, C., Wang, X., Bai, D., and Zhang, H. (2008). Fast svm incremental learning based on the convex hulls algorithm. In *Computational Intelligence and Security, 2008. CIS '08. International Conference on*, volume 1, pages 249–252.

Yacoob, Y. and Black, M. J. (1998). Parameterized modeling and recognition of activities. In *Computer Vision, 1998. Sixth International Conference on*, pages 120–127. IEEE.

Yamato, J., Ohya, J., and Ishii, K. (1992). 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. IEEE.

Yilmaz, A. and Shah, M. (2005). Actions sketch: A novel action representation. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 984–989. IEEE.

Yu, E. and Aggarwal, J. K. (2006). Detection of fence climbing from monocular video. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 375–378. IEEE.

Yu, H., Yang, J., and Han, J. (2003). Classifying large data sets using svms with hierarchical clusters. In *Proceedings of the Ninth ACM SIGKDD International*

Conference on Knowledge Discovery and Data Mining, KDD '03, pages 306–315, New York, NY, USA. ACM.

Yuan, J., Liu, Z., and Wu, Y. (2009). Discriminative subvolume search for efficient action detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2442–2449. IEEE.

Zelnik-Manor, L. and Irani, M. (2001). Event-based analysis of video. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–123. IEEE.

Zhang, D., Gatica-Perez, D., Bengio, S., McCowan, I. A., and Lathoud, G. (2004). Modeling individual and group actions in meetings with layered hmms. In *IEEE Transaction on Multimedia, June, 2006*, number LIDIAP-CONF-2004-002.

Zhang, J., Marszalek, M., Lazebnik, S., and Schmid, C. (2006). Local features and kernels for classification of texture and object categories: A comprehensive study. In *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, CVPRW '06, pages 13–, Washington, DC, USA. IEEE Computer Society.

Zhang, X., Zhang, H., and Cao, X. (2012). Action recognition based on spatial-temporal pyramid sparse coding. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 1455–1458.

Zheng, J., Shen, F., Fan, H., and Zhao, J. (2012). An online incremental learning support vector machine for large-scale data. *Neural Computing and Applications*, 22(5):1023–1035.

Zhou, Z.-H. and Chen, Z.-Q. (2002). Hybrid decision tree. *Knowledge-Based Systems*, 15(8):515 – 528.