Universidad Carlos III de Madrid

Escuela Politécnica Superior

Departamento de Informática

Doctorado en Ciencia y Tecnología Informática

Tesis Doctoral

Action Recognition in Visual Sensor Networks: A Data Fusion Perspective

Rodrigo Cilla Ugarte

Dirigida por

Miguel Ángel Patricio Guisado

Antonio Berlanga de Jesús

December 4, 2012

Web page: http://www.giaa.inf.uc3m.es/miembros/rodri

E-mail: rodri.cilla@gmail.com

Address:

Grupo de Inteligencia Artificial Aplicada
Departamento de Informática
Universidad Carlos III de Madrid
Av. de la Universidad Carlos III, 22
Colmenarejo 28270 — Spain

Action Recognition in Visual Sensor Networks: A Data Fusion Perspective

Autor: Rodrigo Cilla Ugarte
Directores: Miguel A. Patricio
Antonio Berlanga

Firma del Tribunal Calificador:

	Nombre y Apellidos	Firma
Presidente: D.	.....	.....
Vocal: D.	.....	.....
Secretario: D.	.....	.....

Calificación: .....

Colmenarejo,..... de ..... de 2012.

To the PhD students in Spain that will finish their dissertations without funding after the government cuts and to the ones that will have to go overseas to continue their careers. We'll come back! #sinciencianohayfuturo

Contents

List of Figures

List of Tables

Abstract

VISUAL Sensor Networks have emerged as a new technology to bring computer vision algorithms to the real world. However, they impose restrictions in the computational resources and bandwidth available to solve target problems. This thesis is concerned with the definition of new efficient algorithms to perform Human Action Recognition with Visual Sensor Networks.

Human Action Recognition systems apply sequence modelling methods to integrate the temporal sensor measurements available. Among sequence modelling methods, the Hidden Conditional Random Field has shown a great performance in sequence classification tasks, outperforming many other methods. However, a parameter estimation procedure has not been proposed with feature and model selection properties. This thesis fills this lack proposing a new objective function to optimize during training. The L2 regularizer employed in the standard objective function is replaced by an overlapping group-L1 regularizer that produces feature and model selection effects in the optima. A gradient-based search strategy is proposed to find the optimal parameters of the objective function. Experimental evidence shows that Hidden Conditional Random Fields with their parameters estimated employing the proposed method have a higher predictive accuracy than those estimated with the standard method, with an smaller inference cost.

This thesis also deals with the problem of human action recognition from multiple cameras, with the focus on reducing the amount of network bandwidth required. A multiple view dimensionality reduction framework is developed to obtain similar low dimensional representation for the motion descriptors extracted from multiple cameras. An alternative is proposed predicting the action class locally at each camera with the motion descriptors extracted from each view and integrating the different

action decisions to make a global decision on the action performed. The reported experiments show that the proposed framework has a predictive performance similar to 3D state of the art methods, but with a lower computational complexity and lower bandwidth requirements.

Abstract

LAS Redes de Sensores Visuales son una nueva tecnología que permite el despliegue de algoritmos de visión por computador en el mundo real. Sin embargo, estas imponen restricciones en los recursos de computo y de ancho de banda disponibles para la resolución del problema en cuestión. Esta tesis tiene por objeto la definición de nuevos algoritmos con los que realizar reconocimiento de actividades humanas en redes de sensores visuales, teniendo en cuenta las restricciones planteadas.

Los sistemas de reconocimiento de acciones aplican métodos de modelado de secuencias para la integración de las medidas temporales proporcionadas por los sensores. Entre los modelos para el modelado de secuencias, el *Hidden Conditional Random Field* ha mostrado un gran rendimiento en la clasificación de secuencias, superando a otros métodos existentes. Sin embargo, no se ha definido un procedimiento para la estimación de sus parámetros que incluya selección de atributos y selección de modelo. Esta tesis tiene por objeto cubrir esta carencia proponiendo una nueva función objetivo que optimizar en la estimación de los parámetros óptimos. El regularizador L2 empleado en la función objetivo estandar va a ser remplazado por un regularizador grupo-L1 solapado que va a producir los efectos de selección de modelo y atributos deseados. Se va a proponer una estrategia de búsqueda con la que obtener el valor óptimo de estos parámetros. Los experimentos realizados muestran que los modelos estimados utilizando la función objetivo propuesta tienen un mayor poder de predicción, reduciendo al mismo tiempo el coste computacional de la inferencia.

Esta tesis también trata el problema del reconocimiento de acciones humanas empleando multiples cámaras, centrándonose en reducir la cantidad de ancho de

banda requerida par el proceso. Para ello se propone un nueva estructura en la que definir algoritmos de reducción de dimensionalidad para datos definidos en multiples vistas. Mediante su aplicación se obtienen representaciones de baja dimensionalidad similares para los descriptores de movimiento calculados en cada una de las cámaras. También se propone un método alternativo basado en la predicción de la acción realizada con los descriptores obtenidos en cada una de las cámaras, para luego combinar las diferentes predicciones en una global. La experimentación realizada muestra que estos métodos tienen una eficacia similar a la alcanzada por los métodos existentes basados en reconstrucción 3D, pero con una menor complejidad computacional y un menor uso de la red.

Agradecimientos

QUIERO comenzar estos agradecimientos con Miguel Ángel Patricio Guisado y Antonio Berlanga de Jesús, que han guiado todo el trabajo que he venido desarrollando durante los últimos años en el Grupo de Inteligencia Artificial Aplicada de la Universidad Carlos III de Madrid y que culmina con la escritura de esta tesis doctoral. Les deseo que los futuros estudiantes de doctorado que tengan sean mucho mejores que yo, que les hagan mucho más caso y que les lleven la contraria una parte de lo que yo lo he hecho. A José Manuel Molina le tengo que agradecer, entre otras cosas que es mejor que no queden por escrito, que no me haya despedido a pesar de los múltiples motivos que le he dado para ello. Me fichó pensando que iba a ser Pau Gasol pero resulté ser un Epi cualquiera. A Jesús García Herrero y Javier Carbó Rubiera les tengo que agradecer las múltiples discusiones sobre inteligencia artifical que hemos tenido durantes estos años y que probablemente se vean aquí reflejadas de alguna forma.

Con mucha gente he compartido fatigas, marrones, robos y viajes a congresos durante estos años: Vero, Fede, Anita, Gonza, Kike, Mirren, Morlis, Luis, Sandra, Jorge, Luismi, Ramón, Eli, Albertito, José Daniel, Laura, Mar, Cesar, Oscar, Juan... La vida en la UC3M no habría sido lo mismo sin vosotros. Habría tardado un par de años menos en escribir esta tesis y tendría la tensión más baja al haber tomado muchos menos cafés, pero me lo habría pasado infinitamente peor.

Y por supuesto a mis padres. A mi madre por haberme picado para ver quien terminaba antes, carrera que he perdido por otra parte. A mi padre por decirme tantas veces que me busque un trabajo que me dé dinero, para no hacerlo y terminar esto sólo por llevarle la contraria. Y por supuesto a mi abuela, por darme de comer

tantos sábados y domingos en los que no tenía tiempo (ni ganas, para que nos vamos a engañar) de hacerme la comida. También por su inmejorable habilidad para arreglarme los rotos de los pantalones.

También quiero acordarme de todas esas personas a las que me he ido encontrando por el mundo y que han hecho que me sienta menos lejos de casa: Hanna, Anne, Tom, Emilie, Felipe, Luismi, Cande, Guille, Maria, Ainara, Carla, Iván, Bea,... nos vemos en la próxima estancia en Boston, San Francisco o cualquier otro lguar.

Y a todos aquellos colegas que han estado ahí esperándome para tomar cañas al salir de trabajar y a los que tantas veces he dejado tirados en la misión: Moncuar, Luna, Pajares, Ceci, Peibol, Miguelín, Gaby, Fede, Sevilla, Panetone, Isa, Esperón, Mar, Antonio, Pili, Ale, Pelala, Pusi, Vegetal, Enano o Sacarino. ... Sin vosotros el mundo sería un lugar mucho más aburrido.

Rodrigo Cilla Ugarte

Madrid, Diciembre de 2012.

1

Introduction

Big Brother is Watching You

1984. George Orwell

In 1966 Marvin Minsky, an Artificial Intelligence pioneer at Massachusetts Institute of Technology, asked an undergraduate student to solve "the problem of computer vision" as a summer project. Of course he did not solve it, but a new research discipline was born from that challenge. Almost fifty years have passed and a wide variety of visual perception problems have been studied. Nowadays computers recognize objects from their visual appearance, robots navigate in unknown environments employing visual information, visual surveillance systems recognize faces of wanted people and automatic quality control systems discover imperfections in production lines. Automatic video analysis goes one step beyond, employing image streams to study phenomena with a temporal extent. Video trackers measure speed and location of moving objects, enabling the construction of a diversity of systems to automate video surveillance tasks or perform human computer interaction. Despite the advances made in the area, there are multiple problems not yet solved. One of them is the accurate recognition of human movements.

The recognition of human movements (Aggarwal & Ryoo, 2011) has been studied by the computer vision community for more than twenty years. The developments made during this period have enabled the creation of multiple systems. Automatic Surveillance, Ambient Intelligence or Human Computer Interaction are some

of them. Abnormal behavior detection is employed in Video Surveillance Systems to detect suspicious behaviors that might be assessed as a thread. Smart home environments analyze actions and mood of the inhabitants to adapt the environment to their preferences, changing music or lighting conditions to make it more comfortable. Commercial gaming platforms employ advanced sensors to capture the real movements of the players, providing an enhanced and more realistic experience.

However, there are still multiple challenges to be solved before moving human action recognition technologies form controlled laboratories to the real world.

1.1 Visual Sensor Networks

For many years networks of cameras have acquired images and sent them to a central location where they are analyzed. The usage of computer vision systems has displaced humans from analyzing the captured images. In any case all the processing has been located in a centralized system with high computing capabilities and employing wide-band networks for image transmission. These deployments have a high monetary cost, requiring a considerable inversion for their installation and maintenance.

The advances made in imaging sensor technologies, network communications and embedding computing capabilities have led to the creation of networks of smart camera devices, called Visual Sensor Networks (VSNs) (Charfi *et al.* , 2009; Soro & Heinzelman, 2009; Tavli *et al.* , 2011). A VSN consist of a set of camera nodes integrating an imaging sensor, and embedded processor and a wireless transceiver. VSNs constitute a low-cost alternative to the traditional centralized systems in order to foster the application of computer vision in the real world.

A VSN is a distributed system to monitor the environment where it is deployed. Camera nodes process image data locally to extract relevant information and cooperate between them to solve a given task. A central node usually collects the

information generated by the camera nodes in order to present it to the user.

The number of potential application domains benefited from developments in VSN technology ranges from video-surveillance systems to elderly monitoring environments to telepresence. Any application domain where multidimensional information is collected from multiple sensors and processed to infer the state of an entity of the real world is a candidate for the usage of VSNs.

Visual sensor networks might be considered as a second generation of Wireless Sensor Networks (WSNs), augmenting their low sensing capabilities. The kind of sensors employed in WSNs is reduced to scalar sensors measuring pressure, temperature, humidity or presence. With VSNs the sensing capabilities of sensor networks are considerably increased, as rich multidimensional information about the world is now obtained. Unfortunately, the usage of multidimensional information is not free, as higher computational capabilities are required at the camera nodes. The information to send over the - narrow - network is also importantly increased.

However, the usage of standard computer vision algorithms in VSNs is limited by the nature of the networks. The low computational performance and memory capabilities of the embedded processors at the camera nodes restricts the usage of state of the art methods with high accuracy but demanding a lot of computational resources. But beyond the availability of computational resources, what restricts even more the family of candidate methods to employ is the energy consumption restrictions. Camera nodes in VSNs usually have a battery as power supply. Battery life maximization is not compatible with a high computational demand from the algorithms employed.

Other design constraint in VSNs is imposed by the network bandwidth available. For example, the power-efficient ZigBee wireless standard commonly employed in VSNs has a maximum transmission rate of 250 kilobits per second that forbids the streaming of large image data at an acceptable frame rate. Thus, data has to be processed at the camera nodes to minimize the amount of information needed to be sent over the network. But this restriction gets in conflict with the energy and

computational resources restrictions. Fewer data sent over the network probably means higher energy consumption caused by computations, and sending more data over the network also means a higher energy consumption due to transmission.

New computer vision algorithms need to be developed taking into account the design constraints imposed by VSNs, as the focus of computer vision research has been traditionally on accuracy and not in efficiency. They have to minimize their memory usage and the number of operations required to obtain a meaningful result. New data fusion paradigms for visual data are needed too, employing more abstract data with a probably smaller size at the input.

1.2 Human Action Recognition and Visual Sensor Networks

The usage of multiple camera viewpoints in the characterization of human motions has diverse advantages. It allows to construct systems robust against partial or even full body occlusions produced by furniture or walls, something essential when the systems should be deployed in real uncontrolled environments. Other advantage is that the descriptors extracted from the multiple cameras provide complementary information about the performed motion, allowing the creation of systems with higher predictive accuracy. However, current proposals for human action recognition from multiple cameras are not well posed for their deployment in VSN infrastructure.

Traditional multi-camera human action recognition methods are based in the projection of 2D descriptors extracted from each camera into a single 3D descriptor, employing visual geometry information. They need to send to a central node high dimensional descriptors such silhouettes or optical flow fields requiring a lot of bandwidth not available in a VSN. The usage of visual geometry requires to perform calibration of the cameras to obtain the projection matrices. Each time a camera is moved, maybe accidentally, the camera should be recalibrated, complicating the us-

age of VSNs in real uncontrolled environments for the recognition of human actions.

These approaches require streaming a large amount of data in the form of raw images or, at least, silhouette masks containing the objects of interest in the scene. Definitely they send much more information than the acceptable in a VSN, because the local image processing made at the camera nodes is very simple. Thus, a paradigm shift in how the information is processed for multi-camera human action recognition systems is needed, moving processing from the central server to the camera nodes to obtain higher levelinformation more easy to send over the network.

The Data Fusion community has studied the problem of combining measurements from multiple sensors for a long time. The concepts and algorithms developed by them might be applied to guide the development of this new paradigm for the recognition of human actions from multiple cameras.

1.3 Sequence classification, Human Action Recognition and Visual Sensor Networks

A key component of almost any human action recognition system is the sequence modeling method employed. Human motions not happen isolated, they have a temporal extent. Thus, sequence modeling methods have a great importance in the recognition of human actions. Multiple alternatives have been proposed to model the temporal correlations among the visual features extracted from the images sequences containing human actions. They will be in depth reviewed later in chapter 2.

A popular class of sequence models employed in the recognition of human actions is based on probabilistic graphical models. Among them, generative models based on the Hidden Markov Model have become the *de facto* standard model. Multiple works have employed them and proposed variations to capture the particularities of the different recognition scenarios. Efficient exact and approximate

algorithms exist to perform the associated inference tasks. Recently, discriminative sequence models such the Hidden Conditional Random Field (HCRF) (Quattoni *et al.* , 2007) have emerged as a promising alternative to generative models. They have shown a higher predictive performance in different tasks than their generative counterparts. However, they still have a reduced applicability spectrum and they do not have displaced generative models.

This work wants to foster the spread of discriminative sequence models incorporating model and feature selection in the training algorithm of the HCRF sequence classifier. Model and feature selection are two desirable properties in any machine learning algorithm. The Occam's Razor principle of machine learning stands that a model should not be more complex than strictly required. Model and feature selection are two ways of implementing this principle, both reducing the complexity of the estimated model. Model selection in the HCRF refers to the selection of the optimal number of hidden state variables, while feature selection refers to the selection of informative features in the input sequences discarding uninformative ones.

Regarding VSNs, it is desirable to accomplish with the Occam's Razor principle. As already mentioned, the embedded processing capabilities at the camera nodes are restricted due to energy consumption limitations. Then, simplifying the models employed as much as possible - but not more - will lead to accurate algorithms with an smaller computational complexity and consuming an smaller amount of energy. The "Green" buzzword probably might be applied to these algorithms.

1.4 Thesis objectives

This thesis is mainly concerned with the open issues discussed above; that is:

- The efficient estimation of the parameters of the Hidden Conditional Random Field sequence classifier adjusting model complexity while maximizing the predictive accuracy.

- The efficient combination of multiple camera information for the prediction of human actions, reducing the amount of information to be sent from the camera nodes to the central server.

To accomplish the first issue a modification of the regularization strategy employed in the estimation of optimal parameters if the HCRF is proposed. However, the search strategy employed by the standard model is no longer valid and different algorithms will have to be employed for the task.

Two different strategies are going to be analyzed for the efficient combination of multiple camera information, operating at different levels of abstraction. The first combines visual features extracted from each one of the cameras. To this end a new dimensionality reduction framework from multiple views is going to be developed. The framework abstracts different already existing multiple view dimensionality reduction algorithms, and new algorithms are going to be developed under its support. Methods to obtain approximate solutions of the framework in large scale learning are discussed. The second proposal is going to bring the action prediction to the camera nodes. The predictions made by the camera nodes are going to be combined to make a global action recognition decision. A probabilistic formulation is going to be employed to achieve this objective.

All the proposed algorithms are going to be validated employing standard datasets for human action recognition. Their performance is going to be compared to other state of the art methods.

In order to give an strong background supporting this proposals a review of state of the art methods for human action recognition is provided, covering the different steps involved in the process.

Regarding the problem of human action recognition from multiple cameras it is going to be reviewed from the viewpoint of data fusion, employing the concepts and frameworks developed by that community.

1.5 Structure of the thesis

This document is structured into two conceptual blocks, the first one reviewing and categorizing existing methods for Human Action Recognition and the second presenting the proposals of this dissertation.

Chapter 2 presents a review of Human Action Recognition algorithms with the focus in methods employing a single camera for action prediction. A review of previous surveys of the field published along the years is presented to give the reader the idea of how the field has evolved. The different semantic definitions proposed to categorize human motion analysis systems are merged into a new one summarizing all the proposed concepts. Then, different existing proposals to perform feature extraction are reviewed. The chapter finishes describing methods to predict action class labels from the extracted features.

Chapter 3 reviews multi-camera human action recognition systems employing the concepts and frameworks developed by the data fusion community. The JDL process model and Dasarathy's Input-Output model are presented. Works in multi-camera human action recognition are categorized according to them.

In Chapter 4 the exposition of the proposals of this dissertation begins with the model and feature selection procedures for the HCRF. First, the HCRF standard model is presented with inference and learning algorithms. Then, the proposed training procedure is presented with the optimal parameter search strategy to be employed. The chapter finishes with experimental validation of the proposed methods in the recognition of the Weizmann dataset.

Chapter 5 presents an extension of the graph embedding dimensionality reduction framework to the case when multiple views of the data are defined. The chapter begins introducing the standard graph embedding framework and how some well-known dimensionality reduction algorithms are instantiations of it. The chapter follows presenting the proposed extension to the graph embedding framework. Different methods to obtain approximate solutions in large scale scenarios are presented.

The chapter finishes with the validation of the framework in the prediction of the IXMAS dataset.

Chapter 6 presents and alternative model to perform human action recognition from multiple cameras. The single camera recognition method employed to process the streams from each camera is presented. Then, some alternative rules to combine the local predicions are discussed. Experimental evidence is reported applying the proposed system to the prediction of IXMAS dataset showing a similar performance to the method proposed in chapter 5.

Finally, chapter 7 presents the conclusions achieved after the realization of this PhD dissertation, discussing future works with their origin on what has been presented here. A number of complementary appendixes are included mainly for reference purposes. In particular,

Appendix A lists the publications related to this work.

Appendix B describes the datasets and evaluation protocols employed to test the different algorithms

Appendix C presents the auxiliary feature extraction methods employed in the experimental evaluation of the proposed methods.

Appendix D presents the dynamic time warping distance nearest neighbor sequence classifier employed to predict actions in the experiments presented in chapter 5.

For further details regarding this thesis and its related publications please visit the author's web page (http://www.giaa.inf.uc3m.es/miembros/rodri)

I

State of the art

2

Human Action Recognition from Video

> If you know others and know yourself, you will not be imperiled in a hundred battles;
> if you do not know others but know yourself, you win one and lose one;
> if you do not know others and do not know yourself, you will be imperiled in every single battle.
>
> The Art of War. Sun Tzu

THIS chapter presents the advances made in the field of Human Action Recognition from video during the last two decades, with an special emphasis on feature extraction algorithms and action recognition models. The general steps to perform action recognition are introduced in first term. An analysis of the previous published surveys of the area is presented with the aim of reviewing how the research trends have evolved over time. Then, a discussion about the meaning of different terms such movement, action or activity and how have been employed in the literature is presented, with the aim of fixing their usage in further chapters. After presenting these general topics, feature extraction methods and action recognition models are in depth reviewed. Chapter finishes with a critical discussion about the achievements reached.

2.1 Action Recognition Process

Multiple proposals have been developed with the common objective of bridging the semantic gap between pixel intensity values at the input sequences and high level descriptions about their content. Every work defines its own steps to incrementally perform the transformation but, in general they might be grouped in two categories:

1. **Feature extraction**. The objective of this step is the extraction of informative attributes about the motion of interest in the video. Ideally, the attributes should be invariant towards changes in scene illumination, anthropometry or camera viewpoint to make further steps easier. High dimensional data analysis methods are employed to provide compact representations of the features to prevent the *course of dimensionality* in further steps.

2. **Action modeling**. At this step, the features previously computed are transformed into semantic representations. Spatio-temporal correlations among the extracted features and action labels are modeled to build predictive models for the actions of interest.

This division will be employed later in this chapter to organize in first instance the different proposals related to human action recognition, including them in the category where they make their main contribution.

Related to action recognition is multiple object tracking (MTT), as the location and temporal labelling of the objects of interest in the input video is usually a prerequisite in some applications. MTT is out of the scope of this work. Readers are referred to the specific surveys about MTT for further details (Yilmaz *et al.* , 2006).

2.2 Previous surveys

The field of Human Action Recognition has been surveyed by different authors in the last twenty years. A global perspective of the research direction explored during

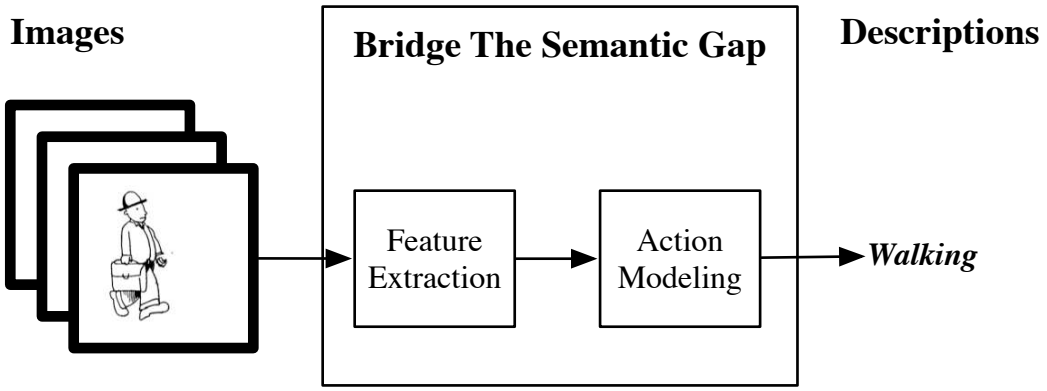


Figure 2.1: A general action recognition pipeline

these years might be obtained if these surveys are analyzed in sequential order.

The first survey was published back in 1995 when Cédras and Shah collected the works of the emergent new topic (Cédras & Shah, 1995). They identified different application areas that would be benefited from the future advances. They reflect the first usages of motion descriptors computed from trajectories, optical flow or silhouettes, with the first models and heuristics for motion classification. Preliminary proposals of tracking methods for human motions were also presented.

In 1999 Gavrila published a survey centered on the representation and tracking of the human body(Gavrila, 1999). He distinguishes between proposals in 2D without explicit shape models, in 2D with explicit shape models and in 3D. Advances in action models were presented also. He identified some challenges that would have to be solved in future works, such the difficulty of acquiring 3D models in uncontrolled situations or the difficulties that current systems have to handle occlusions. The lack of ground truth data is presented as an issue for the fair comparison of existing approaches. That should be addressed in subsequent works. Finally, he postulates the usage of 3D range data as a way to improve the acquisition of human models.

The same year Aggarwal and Cai published a complementary survey (Aggarwal & Cai, 1999) where a lot of importance is given to human body tracking methods. A first categorization for the action models is proposed, differentiating between tem-

plate matching approaches and state-space based approaches.

Moeslund and Granum published their first survey in Human Motion Capture in 2001 (Moeslund & Granum, 2001). They identify 4 steps in the analysis of human motion: initialization, tracking, pose estimation and recognition. The survey is organized according to these steps. They realize that although most of the methods proposed until then are based on human body models in 2D or 3D, recent works have shown promising results with model-free approaches. These methods will become very popular in subsequent years as they not depend on a good human model acquisition strategy.

Hu et al. presented in 2004 a survey from the viewpoint of video surveillance applications (Hu *et al.* , 2004), presenting works in object detection, identification, tracking and abnormal behavior detection. Particular attention is given to the fusion of data obtained from multiple cameras. The need for systems robust towards occlusions is again identified. They suggest the need to go beyond behavior recognition and build systems to predict future behaviors in advance.

In 2006 Moeslund et al. published a new version of their previous survey (Moeslund *et al.* , 2006). Recent developments in the areas previously covered are presented, including advances in model-free methods. A shift in the human action recognition paradigm has started, progressively leaving model based approaches in favour of model-free approaches. As an issue to address in future works they present the lack of high level behaviour models for the recognition of complex activities, as most of the works presented until then have been centered in the deep study of the details of simple actions.

Turaga et al. publish their survey in 2008 (Turaga *et al.* , 2008), and for the first time knowledge and logic-based approaches for the recognition of complex human actions are presented. They identify as topics to be addressed in the future the adaption of the systems for real-world operation, the creation of human action recognition systems view, rate and anthropometry invariant, the exhaustive evaluation of the systems, the integration with other sensing modalities and, again, the reasoning about

the future intentions of people.

Poppe makes an extensive review of the methods for feature extraction and recognition of simple actions (Poppe, 2010). He includes different datasets for the evaluation of human action recognition methods published in recent years, and warns about the need for a multidataset evaluation framework, as evaluated methods might be biased towards some specific dataset. Furthermore, the need for application specific datasets with specialized evaluation metrics is identified, as different application have different operational requirements.

The survey published by Weinland et al. gives a special attention to novel works in view-invariant action recognition and temporal segmentation of actions (Weinland *et al.* , 2011). He emphasizes the need for challenging action recognition datasets in uncontrolled scenarios with the actions having a wide variability. Methods based in local feature extraction have shown promising result in such scenarios and are identified as a target of study as there is a lot of open problems related to them.

Aggarwal and Ryoo recently published a new survey (Aggarwal & Ryoo, 2011), organizing the works according to the complexity of the recognized actions. An special attention is given to group activity recognition and human-object interaction. They point out that although a big progress has been made in recent years in the domain, the problem is far from being solved as still there are a lot of practical issues to be addressed. Efficient algorithms for real-time operation, the ability to recover from tracking errors or the recognition of complex activities by spacio-temporal features are identified as future research lines.

A review of these surveys shows a big progress in the field of human action recognition in the last two decades. The seminal systems proposed in the mid-90's recognizing basic gestures have evolved to study complex activities performed by multiple people. Action Recognition is performed with *videos in the wild* i.e., sequences acquired in uncontrolled environments where occlusions and background noise increase the difficulty of defining robust models. Models invariant towards camera viewpoint changes and anthropometries have been defined, augmenting the

robustness of the systems towards changes in the environment and enabling the transferability of models. However, the action recognition problem is still far from being solved, as multiple opened problems of computer vision and artificial intelligence affect the field. At the end of this chapter some of them will be discussed.

2.3 Motions, Events, Actions, Activities and all that

Hierarchy	**Categories**
(Nagel, 1988)	*change*, *event*, *verb*, *history*
(Bobick, 1997)	*movement*, *activity*, *action*
(Bremond & Nevatia, 2000)	*simple events*, *composed events*, *multi-threaded events*
(Aggarwal & Ryoo, 2011)	*gestures*, *actions*, *interactions*, *group activities*
Proposed	*Gesture*, *Action*, *Individual activity*, *Interactions*, *Group activities*

Table 2.1: Different action hierarchies

The first question that must be answered when facing a human motion analysis problem is related to the semantic definition of the motions in consideration. Different motions have different complexities. Imagine an scenario when a person waves his arms. The waving movement is composed of two different motions. First, the person rises his arms. Then, the person lows his arms. The waving motion is composed of the temporal concatenation of the simple motions rise and low. Complex Motions are composed by the concatenation of simpler motions. At the same time, the motion might have a meaning depending on the context where the motion is performed. Even, the same motion can have different meanings in different contexts. For example, a person waves his hands in a station to signal somebody that is there. By contrast, when an airport marshal waves his hands to a plane, he wants to signal the plane to stop. Motions are almost the same, but the meanings are different. Two different abstractions have been shown related to this sample waving actions, one related to the motions and the other related to the meaning. It is clear that some

hierarchical structure exists related to the motions, and the problem of defining the different levels of the motion hierarchy has been tackled by different authors, without achieving a common consensus.

Nagel, in a first attempt to categorize the action hierarchies in video, defined the concepts *change*, *event*, *verb* and *history* (Nagel, 1988). Bobick defines three different semantic levels related to the amount of knowledge required to classify each one of them (Bobick, 1997). He defines a *movement* as "a motion whose execution is consistent and easily characterized by a definite space-time trajectory in some configuration space". An *activity* is defined as " a statistical sequence of movements". Finally, an *action* is defined as an activity performed in some particular context.

Hongeng et al. (Bremond & Nevatia, 2000) proposed a event hierarchy composed of *simple events*, *composed events* and *multithreaded events*, allowing the modelling of motions where more than one entity is involved. *Simple events* are defined as coherent short motions described by a set of logical restrictions imposed over a set of *sub-events* or directly over a set of target properties. *Composed events* are defined as temporal sequences of *simple events* or other *composed events* sequentially ordered along time. *Multithreaded events* are defined imposing logical and temporal restrictions over two or more events executed in parallel, typically to model events involving multiple entities.

Aggarwal and Ryoo (Aggarwal & Ryoo, 2011) define four levels of abstraction, named *gestures*, *actions*, *interactions* and *group activities*. *Gestures* are defined as elementary movements of a person's body part, and are the atomic components describing the meaningful motion of a person. *Actions* are defined as single-person activities that may be composed of multiple gestures organized temporally. Interactions are human activities that involve two or more persons and/or objects. *Group activities* are defined as the activities performed by conceptual groups composed of multiple persons and/or objects. The inclusion of the former category is novel with respect to previous taxonomies.

No action hierarchy is generally accepted, and the works in Human Action Recognition employ terms such *event*, *motion*, *action* or *activity* without a clear common meaning. The only fact generally accepted is that motions are organized in hierarchies, where complex motion concepts are composed of simpler ones. Although this lack of common definitions might be understood as a handicap to understand the works in Human Action Recognition, in practice it is not as the different works have a similar structure. This work contributes to the confusion of terms defining five levels of abstraction mixing the concepts formulated on previous categorizations.

1. A *Gesture* is defined as the elementary movement of a person's body part, and are considered as the atoms of motion analysis.

2. An *action* is built defining spatial and temporal restrictions over a set of *gestures*. Examples of actions are "running","walking" or "waving".

3. *Individual activities*, or simply, *activities* are defined as *actions* performed in a given context.

4. *Interactions* are defined as actions where more than one entity is involved. Interactions might be defined between humans or between humans and objects.

5. *Group activities* are defined as the motions performed by conceptual groups composed of multiple persons and/or objects.

This hierarchy is similar to the defined by Aggarwal and Ryoo (Aggarwal & Ryoo, 2011), splitting their definition of *action* into *actions* and *individual activities* according to the level of the knowledge required at each level, in the style of Bobick (Bobick, 1997).

The works examined in this chapter and the proposals of this dissertation are related to the recognition of *gestures* and *actions*, not considering higher levels where the meaning of the performed motions depends on scene knowledge.

2.4 Feature extraction

The first step in the Human Action Recognition chain is to select the proper cues to describe the actions of interests. Ideally, the selected cues have to capture the variance of the target motion while at the same time have to be robust towards the noise in the images, scene illumination, occlusions and changes in the camera viewpoint. There is no cue valid for every action recognition task.

The different features proposed for human action recognition are going to be divided in three different categories:

- Model-based features, employing the parameters of a model representing target properties.
- Global features, extracting visual information from the image region occupied by the target.
- Local features, extracting visual information about local changes in the image cause by the motion of the target.

Next paragraphs will review the particularities of these approaches and present relevant works. Feature encoding methods are presented as a complement, as their usage is widely spread to prevent the *course of dimensionality* and visualize the extracted features.

2.4.1 Model-based features

The first group of features discussed here are obtained from a model fitted to the entity under analysis. The model abstracts the shape of the entity, representing it on some parametric space. The abstraction might be performed at different levels of detail, depending of the quality of the observations of the target and the computational resources available. Rough models are easier to compute than detailed models, at

the cost of capturing less accurate information about the performed motion. The object tracking survey of Yilmaz et al. (Yilmaz *et al.* , 2006) presents a variety of models for the entities and techniques for the estimation of the optimal parameters.

At the lowest level of detail the entities are represented by a single point. This is typical in far-field views where the entities are very small and there is not enough detail in the representation to try any other approach more sophisticated. Inferences about motion at this level are limited to reasoning about the speed of the target and the trajectory type.

At the mid level of detail the entities are represented by a rectangular bounding box, incorporating the spatial extent of the entity into the model. Features such the bounding box width, height, aspect ratio or rotation are included in the model. This enables performing simple inferences about the kind of motion and pose of people, i.e. if it is walking, running or stopped.

At the highest level of detail the entities are represented by part based models, where the different articulated parts of the entity are independently modeled to provide an accurate representation of the entity shape. In the case of humans these corresponds to the body limbs. They are represented either as segments or closed surfaces and approaches have been proposed both in 2D and 3D. Marr and Nishihara introduced a hierarchical cylinder model to represent the human body in 3D (Marr & Nishihara, 1978). Different levels of detail are defined decomposing the cylinder containing the full human body into subcylinders containing the limbs. The limbs are also decomposed into subcylinders to represents the different parts and joints in the limbs. A more refined model might be built employing super-quadrics instead of cylinders (Gavrila & Davis, 1995). The main problem of this models is that the estimation of model parameters is very challenging because of the high number of degrees of freedom in the models. The representation of the body limbs in 2D is performed in a similar way, but with 2D primitives. Park and Aggarwal represent the human body as a set of ellipses and convex hulls (Park & Aggarwal, 2004). Fanti et al. proposes a probabilistic model to find the configuration of the body parts (Fanti

et al. , 2005).

Temporal moments and derivatives of the model parameters are employed to build robust features to introduce in the classifiers. A large number of the measurements that might be employed is presented in (Ribeiro & Santos Victor, 2005).

These models allow inferences about detailed gestures and actions of people (grasping, Tango dance steps (Gavrila & Davis, 1995),...). However, the complexity of recovering model parameters has relegated the usage of model based features in favour of global and local image features.

2.4.2 Global Image features

Given the region where an object of interest is located, two different kinds of features might be extracted:

1. Appearance features, representing how a target looks like.
2. Motion features, representing how a target moves.

2.4.2.1 Appearance features

Appearance features describe how the entity under analysis looks like at a given instant. Information about the instant pose of people is captured by this attributes. The variations in body shapes and clothing corrupt pose signal, but in practice is not a real problem as multiple actors are employed for model training.

Silhouette features The temporal evolution of human silhouettes is a powerful cue for the recognition of human actions where the relative movement of body limbs is important, such in human computer interaction applications. Human silhouettes are binary image masks with the pixels belonging to the human activated. The simplest way to obtain them is to subtract a reference background image from the current

frame and threshold the result. There are multiple methods to build background models and perform the extraction of the silhouettes, but they are out of the scope of this survey. Readers are referred to (Piccardi, 2004) for information in the topic.

In a seminal work in human action recognition (Yamato *et al.* , 1992) it was proposed the usage of raw human silhouettes to recognize different tennis strokes. Since then raw human silhouettes, using different coding schemes, have been a recurrent feature to predict actions (Wang & Suter, 2008).

Other works have propose the usage of distance transforms to represent silhouettes (Wang & Suter, 2008). Given a binary image with a silhouette, the value of each pixel is replaced by the distance to the border. The resulting image has higher values at pixels close to the skeleton and lower values at pixels close to the boundary, representing the morphological structure of the input silhouette. Distance transforms have experimentally shown better prediction accuracy than raw silhouettes(Wang & Suter, 2008).

Other transform that has been proposed to code the variations in human silhouettes is the $\mathcal{R} - transform$ (Wang *et al.* , 2007a), an extension of the Radon transform (Helgason, 1999). The transform is invariant to image translation. In addition, rotation and scaling of the input image have predictable effects allowing further normalization.

It is also possible to employ the Discrete Fourier Transform and the Discrete Wavelet Transform to describe the human silhouette, achieving invariance towards rotation, translation and scale after proper normalization (Ragheb *et al.* , 2008)

Space-time silhouettes Any of the appearance features presented until now employ temporal information to represent the variability of the human actions modelled. These works relay the modelling of the temporal correlations of the extracted features to the classifier. Although these approaches have shown to be effective, other authors have proposed features coding the temporal evolution of the appearance.

Temporal templates (Bobick & Davis, 2001) were proposed as a first attempt to encode the temporal evolution of the appearance into feature vectors. At each instant, the Motion Energy Image (MEI) employs a backward search window to look for pixels containing the silhouette in the past, giving to them an active value in the current binary feature vector. The Motion History Image (MHI) replaces the binary values of the MEI with integer values proportional to the time past since a given pixel was in the silhouette. An extension of the MHI is the Pixel Change History (Xiang & Gong, 2006) (PCH). Instead of giving the highest value to a pixel just added to the silhouette, it linearly increments is value along time to mitigate the effects of noisy observations. The MHI has been extended to 3D (Weinland *et al.* , 2006), computing Motion History Volumes (MHV) from visual hulls instead of silhouettes.

Other authors stack 2D silhouettes into 3D action volumes to correlate the temporal evolution of the silhouettes. The surface of the generated volume can be analyzed employing differential geometry (Yilmaz & Shah, 2005), obtaining characteristic points. Others analyze the entire volume (Gorelick *et al.* , 2007), measuring space-time saliency and orientations of the voxels, and global moments of the entire volume. It is possible to define different moments of the volume (Achard *et al.* , 2007). These 3D volumes have been employed to build models of the variations in the descriptors produced by the change of viewpoint (Lewandowski *et al.* , 2010a). Instead of stacking 2D image sequences, it is also possible to stack their $\mathcal{R}-$ *transforms* (Souvenir & Babbs, 2008). Other possibility is to segment the silhouettes in different subregions and make the matching for the different parts (Ke *et al.* , 2007).

The main drawback of silhouette stacking proposals is that they are not suitable for real time applications, as the attribute extraction is made from the entire volume needing data from the future to compute the features at a given instant.

Other appearance descriptors When the targets are not big enough or too noisy it can be very difficult to obtain silhouettes with enough discriminative power to

discern between the actions being performed. To this end different alternatives have been proposed in order to compute appearance descriptors without the requirement of the silhouette. The 2D Histogram of Oriented Gradients descriptor has been used to track and categorize the motions of hockey players (Lu *et al.* , 2009). Gabor filters have been employed to extract features at different scales and rotations (Escobar *et al.* , 2009). Histograms of line orientations have been built from the output of edge detectors (Ikizler *et al.* , 2008). The similarity of two video volumes might be measured employing Tensor Canonical Correlation Analysis, without relying on explicit target location. Grayscale intensities are employed as features, not relying on any other feature extraction algorithm (Kim & Cipolla, 2009).

2.4.2.2 Motion features

An alternative to the usage of appearance information is motion information. Information about the motion of body limbs is captured instead of information about their current pose. This information is orthogonal to appearance, and in fact lot of systems combine both to increase their robustness (Tran & Sorokin, 2008)

Optical Flow The main motion descriptors are based in optical flow. The optical flow is defined as the apparent motion of the pixels in a sequence of images, providing a velocity vector field over the pixel built from the intensity changes. Different methods have been proposed to solve the partial differential equations defining the field. These methods are out of the scope of this survey. The reader can refer to (Baker *et al.* , 2011) for more information about opical flow estimation.

A first approach (Cutler & Turk, 1998) segmented the motion field into motion blobs, obtaining approximately one for each moving limb. Blob properties such size, position or speed are employed as action descriptors.

A popular approach to process optical flow fields is the proposed by Efros et al. (Efros *et al.* , 2003). The vector field is blurred in first term to remove noisy

components. Then, the field is divided into four different channels, according to the sign and direction of each component. A variation of this descriptor has been proposed simply splitting the motion field into the vertical and horizontal channels (Tran & Sorokin, 2008).

An alternative is to compute different global moments of the field, such the mean, deviation and other high order moments (Ribeiro & Santos Victor, 2005). It is also possible to study physical properties such divergence or vorticity from the temporal evolution of the vector field(Ali *et al.* , 2010).

Histograms of Oriented Optical Flow (Chaudhry *et al.* , 2009) have been built quantifying the direction of each component of the flow field and building an histogram of occurrences. This alternative is robust towards scaling of the target.

Recent works have employed 3D optical flows, projecting the flows computed at multiple cameras (Holte *et al.* , 2011a).

Other filters Other kind of filters, usually faster to compute, have been proposed as alternatives to optical flow to encode motion information.Infinite Impulse Response filtering is proposed to detect the boundaries of moving limbs (Masoud & Papanikolopoulos, 2003). The gradient of the motion features computed this way implicitly encodes the speed and direction of local motions. The standard deviation of the intensities in adjacent frames has been also proposed to estimate the amount of motion (Zhang *et al.* , 2008). However, theses filters are only appropriate to encode motions performed at similar executions rates.

3D Discrete Wavelet Transform has been applied to analyze video volumes(Rapantzikos *et al.* , 2007). The highest coefficients in the transform correspond to the moving body parts, and a fixed number of them is selected to build the final descriptor. Diverse configurations of Gabor filter banks have been employed to analyze volumes of adjacent frames (Chomat & Crowley, 2000; Jhuang *et al.* , 2007; Schindler & Gool, 2008), leading to a descriptor containing the orientation patterns of the motion of the analyzed frames.

2.4.3 Local Image Features

The image features presented in previous section depend on the location of the bounding box of the human to track,

An alternative is to search the image sequence for spatio-temporal points with a particular motion and compute a local feature descriptor capturing the structure of the local motion around the point. The main advantage of this features over global image features is the theoretical robustness to occlusions they present, as the modeling of the global motion of the entity is made by the sum of the detected local motions. Other advantage is that they do not need from target localization, being suitable to model "actions in the wild". This techniques are adapted from the successful local representations proposed for object detection and characterization. The usage of this techniques is divided in two different steps. The first step is related to the location of the spatio-temporal salient points to be employed. The second is related to the description of the motion characteristics in the neighborhood of each each detected salient point.

Different alternatives have been proposed to locate salient points in Image Sequences. The first approach to this problem was to extend the 2D Harris corner detector and automatic scale selection to 3D (Laptev, 2005), in order to detect pixels in video sequences with high intensity variations in space and time. Interest points found in this way are located at spatio-temporal corners. However, many motions, such the spinning of a wheel, do not generate spatio-temporal corners. An alternative to corner location is to employ a quadrature pair of Gabor filters defined in the spatial and temporal dimensions (Dollar *et al.* , 2005), detecting the interest points at the maxima of the filter response. This approach detects a wide variety of kinds of motions, including "spatio-temporal corners". However, it is not able to detect pure translational motions as they do not generate a response in the temporal dimension. An alternative is to employ frame differencing to detect motion and then apply Gabor filtering to the detection (Bregonzio, 2009). This way a better detection of

the body parts generating the motion is achieved, discarding spurious detections at highly textured background zones. The Kadir and Brady information theoretic detector (Kadir & Brady, 2001) has been also extended to 3D (Oikonomopoulos *et al.* , 2006), computing a local entropy measure for salient point detection. The determinant of the Hessian has been proposed as an alternative saliency measure (Willems *et al.* , 2008). The usage of Non-negative matrix factorisation has been proposed to represent input video images into spatial subspace images and a temporal coefficient vector (Wong & Cipolla, 2007). Salient points are located in the obtained decomposition. Experimental results have shown that this method outperforms (Dollar *et al.* , 2005) and (Laptev, 2005). A recent approach has removed from the filter employed the temporal dimension. A Harris corner measure is obtained for each pixel on each frame. This measure is corrected to discard regular geometric patterns that are not likely to be part of a human. Local maxima of each frame is obtained as interest points, removing the static interest points.

The representation of the motion characteristics of a spatio-temporal salient point is obtained computing an spatio-temporal descriptor around the neighborhood of the interest point. Gaussian Derivatives and optical flow are common choices of local attributes employed to create the description (Laptev & Lindenberg, 2006). PCA-SIFT like descriptors are presented in (Dollar *et al.* , 2005), where the one based in brightness gradients outperforms the others. The popular 2D SIFT descriptor has been extended to 3D, showing and accuracy higher than gradient based methods (Scovanner, 2007), but is beaten by the 3D Histogram of gradients descriptor (Klaeser *et al.* , 2008). The SURF descriptor has been also extended to 3D (Willems *et al.* , 2008). Other descriptors encode the alignment of a salient point with respect to its neighbors, fitting a B-spline and taking the polynomial derivatives as the output (Oikonomopoulos *et al.* , 2009).

A fair comparison of the performance of the different proposals for action recognition based on local features has been reported (Wang *et al.* , 2009), showing that there is no method outperforming the others, although the combination of gradient

and optical flow in the descriptor seems to be a good choice.

Related to local features, Bag of Word methods for action modelling are going to be discussed on section 2.5.1.3.

2.4.4 Feature Encoding

The curse of dimensionality in machine learning (Hughes, 1968) stands that fitting models in high dimensional spaces is an ill-posed problem, because the volume of the space where the attributes are defined increases exponentially with the number of dimensions. This phenomena increases the number of required samples to accurately fit model parameters, producing overfitting, i.e. generating models with low predictive performance when new instances are presented. Most, if not all, of the action models that will be presented in next section are learned from training samples and most of the action descriptors presented in previous section have a high number of dimensions. This motivates the usage of coding techniques to reduce the number of dimensions of the human action descriptors while preserving the variance they contain about the observed motions.

The most simple and widely used dimensionality reduction method is Principal Component Analysis (PCA) (Pearson, 1901). It finds linear projections of the input data maximizing their variance in the projected space. The solution to PCA is given by the eigendecomposition of the data covariance matrix computed after normalizing the input data. The eigenvector with the highest associated eigenvalue is known as the first principal component and so on. A vector x in a high dimensional space R^d is projected to a low dimensional space R^m by the linear operation $y = Wx$, where $W \in R^{mxn}$ is a projection matrix composed of the concatenation of the m first principal components.

The main drawbacks of PCA is its linear nature and the gaussianity assumption on the input data. Non-linear variations of the input data are not well represented in the projected space. To overcome this limitation it is possible to apply kernel methods to

PCA. Kernel Principal Component Analysis (KPCA) (Schölkopf *et al.* , 1997) is a non linear extension of PCA, reformulating the computation of the covariance matrix in the form on inner products of the input data. The inner products are then replaced by kernel functions. If the kernel functions are non linear the operation is equivalent to perform the inner products of the input data in a very high dimensional -possibly infinite- feature space, where the variations of the data become linear.

KPCA has the side effect of allowing the transformation of non-linear high dimensional data, such histograms or $\mathcal{R} - transforms$, to the Euclidean space employing kernel functions defined in the space where the data is defined. For example, X^2 distance kernels allow the transformation of histograms to the Euclidean space (Chaudhry *et al.* , 2009). Diffusion distance kernel allows the transformation of R R-transforms (Souvenir & Babbs, 2008). This simplifies the visualization of the descriptors and allows the application of standard techniques, defined in the Euclidean space, in further steps.

The main drawback of KPCA, legated from Kernel Methods, is that it requires from solving the eigendecomposition of a matrix of size NxN, being N the number of training samples. By contrast, the complexity of solving PCA scales with the number of input dimensions d, being linear in the number of training samples. It still assumes that the data will be gaussian in the transformed space.

Other non linear alternative to PCA is given by graph based manifold learning methods, not assuming the gaussianity of the data. They assume high dimensional data might be parametrized by a manifold with a few degrees of freedom. The structure of this manifold is modeled constructing neighborhood graphs in the high dimensional space. To obtain the low dimensional manifold parametrizations of the input data, an objective function is optimized to preserve some property of the manifold. The Isomap (Tenenbaum *et al.* , 2000) algorithm finds low dimensional representations of the input data preserving geodesic distances in the manifold between points . Isomap is not well suited to work with high dimensional data structured in clusters, as the approximation of the geodesic distances between data in differ-

ent clusters is not well suited, because the space to traverse between them is empty. To overcome this problem, Laplacian Eigenmaps (Belkin & Niyogi, 2001) find an embedding of the input data such similar examples in the high dimensional space remain close in the low dimensional space, preserving the cluster structure.

The main drawback of Isomap and LE is that they not provide an embedding function, just the low dimensional embedding of the input data. Although it is possible to employ some auxiliary regression method to learn the mapping function, it is also possible to linearize the embedding problem, obtaining a linear mapping function from the high dimensional space to the low dimensional. The linearization of Isomap is called Isometric Projection (IsoP) (Cai *et al.* , 2007a), while the linearization of LE is called Locallity Preserving Projections (LPP) (Niyogi, 2004). In a similar way to PCA, LE and IsoP might be rewritten in terms of inner products, substituting them by non linear kernel functions and obtaining non linear projection functions respectively named Kernel Isometric Projections (KIsoP) and Kernel Locality Preserving Projections (KLPP).

Manifold Learning methods have been applied in different works. Isomap has been employed for the analysis of $\mathcal{R}$-transform surfaces (Souvenir & Babbs, 2008). LPP has been applied for the analysis of raw silhouettes and distance transforms (Wang & Suter, 2008) and radial distance surfaces (Azary & Savakis, 2010) to learn view-invariant action representations. KLPP has been applied to reduce the dimensionality of Fourier and Wavelet descriptors (Wang *et al.* , 2008b) outperforming the performance of KPCA. Temporal extensions of LE and Isomap have been applied to analyze silhouette and optical flow data , improving the performance with respect to the original methods (Lewandowski *et al.* , 2010b).

Other class of methods, instead of trying to find a low dimensional representation of the descriptors, apply discretization to the input data. Each input descriptor is represented by a discrete label. In particular, this strategy is commonly employed with local feature descriptors for the creation of bag of words models. The common strategy is to run some clustering algorithm to obtain prototype descriptors from

training data and represent each instance with the nearest prototype. K-means clustering is a common approach (Tran & Sorokin, 2008). Kohonen Self Organizing maps have been applied to abstract the view and temporal variations of the data (Martinez-Contreras *et al.* , 2009), representing data instances by the index of the winner neuron. Maximum Mutual Information clustering has been applied to learn discriminative codebooks to represent local descriptors (Liu & Shah, 2008), improving predictive performance. Instead of representing each local feature by the nearest prototype, it can be represented by the linear combination of a few atoms employing sparse coding techniques (Zhu *et al.* , 2011).

2.5 Action Modeling

The higher step in Human Action Recognition is to fed the attributes computed in previous sections into an predictive model to infer the desired knowledge about the action. The diversity of motions that have been studied has motivated different inference tasks to be performed. A - possibly incomplete - categorization of the high level inference tasks that might be performed related to an observed motion sequence $X = \{x_1, \dots, x_T\}$ is:

- Classification: This is, possibly, the simplest task. Given the sequence X, the problem to solve is to provide an action label $y \in \{y_0, \dots, y_N\}$ to the whole sequence.
- Segmentation: Given the sequence X, the problem to solve is to provide a set of action labels $Y = \{y_1, \dots, y_T\}$ to each one of the sequence instants.
- Abnormality detection: Given the sequence X, the problem is to decide if it is coherent with a model of the expected motion.

Another categorization of the inference tasks is given by how the temporal reasoning is performed:

- Filtering: Given the sequence of observations $X = \{x_1, \ldots, x_t\}$, the task is to infer y_t, the most likely state at time t.

- Smoothing: Given a sequence of observations $X = \{x_1, \ldots, x_t\}$, the task is to infer the most probable explanation $Y = \{y_1, \ldots, y_T\}$ of all the hidden states generating the sequence. This task is performed offline.

- Fixed lag smoothing: Given the sequence of observations $X = \{x_1, \ldots, x_t\}$, the task is to infer the most probable explanation $Y = \{y_t - T, \ldots, y_t\}$ for the states in a temporal window of length t. It allows the refinement of the hidden state values obtained with filtering.

- Prediction: Given the sequence of observations $X = \{x_1, \ldots, x_t\}$, the task is to infer the hidden state values $y_{t+\delta}$ in future instants, $\delta \geq 1$.

2.5.1 Sequence models

2.5.1.1 Exemplar based

Given a set of sequences, it is possible to compute different distance measurements between them in order to perform classification tasks employing the nearest neighbor classification rule. Given a set of previously observed sequences and their labels, the label for a new observed sequence is given by the - possibly weighted - vote of the labels corresponding to the k sequences minimizing the distance to the test example.

Different distance measures have been proposed to compare sequences. A correlation measure might be defined employing a temporal window around each frame to obtain the most similar frame from the exemplar database (Efros *et al.* , 2003). Majority voting for each frame is employed to perform sequence classification. However, this approach is not appropriate to compare sequences with motions executed at different rates.

Different proposals have been proposed to circumvent this problem. The Dynamic Time Warping algorithm (Vintsyuk, 1968) performs a time alignment an normalization of a pair of sequences of different lengths to provide a distance measure by means of a temporal transformation of the sequences. It has been employed for gesture recognition (Corradini, n.d.) and for matching of silhouette sequences parametrized in manifold spaces (Blackburn & Ribeiro, 2007). Bicubic interpolation has been employed to transform the query sequences to the same length and compute correlation measures (Wang & Suter, 2007). Other possibility shown in the same work is to employ the Haussdorf distance between the sequences, defined as the median of the minimum distances between every pair of frames, without the need of performing temporal scaling of the sequences.

The main drawback of exemplar models is that every test sequence should be matched against the database of exemplars. Approximate matching methods (He *et al.* , 2012) reduce the complexity to the logarithm of the examples in the database, but still this is a high cost for real time applications. In practice their usage is limited to show the efficiency of low level methods as they have a good predictive performance.

2.5.1.2 Graphical Models

Exemplar and Bag of Words models, although effective, have a very narrow applicability, as they are limited to sequence classification. Then, more advanced models have to be defined in order to allow higher level inference tasks such sequence segmentation. Structured machine learning methods allow the modelling of the probability distributions of sets of labels and observations, incorporating temporal correlations between the observed variables, allowing the realization of segmentation tasks.

Generative models Generative models represent the joint probability distribution $P(X, Y) = P(Y)\,P(X \mid Y)$ of a set of observations X and a set of labels Y. The

standard model to perform Human Action Recognition is the Hidden Markov Model (Rabiner, 1989) (HMM). HMM models the joint probability distribution $P(X, Y)$ of a sequence of observations $X = \{x_1 \dots x_T\}$ and the hidden states $Y = \{Y_1 \dots y_T\}$ generating the observations. The meaning of the hidden states Y is different depending on the action recognition task being performed. In a segmentation task they correspond to the action performed at each instant, while in other task they do not have any special meaning and are considered a hidden model parameter.

The HMM assumes the hidden state sequence Y evolves according to a Markov Chain, parametrized by a conditional probability distribution $P(y_t \mid y_{t-1})$, i.e, the value of a hidden state depends on the value at the previous instant. At the same time, each hidden state is the responsible of generating the observation x_1 according to a probability distribution $P(x_t \mid y_t)$. Thus, the joint probability distribution of the hidden state sequence Y and the sequence of observations X is factorized as $P(X, Y) = \sum_{t=0}^{T} P(x_t \mid y_t) P(y_t \mid y_{t-1})$.

Different emission distributions $P(x_t \mid y_t)$ have been employed to allow different encodings of the observations sequences. Bernouilli emissions are employed in the case of discrete observations . Gaussian distributions and Gaussian Mixture Models are employed as conditional probability distribution in Euclidean spaces (Rabiner, 1989). Kernel Density Estimation observation distributions have been proposed to model observations non parametrically (Piccardi & Pérez, 2007).

HMM is the simplest element of the family of generative models known as Dynamic Bayesian Networks (DBNs) (Murphy, 2002). More complex instances of this family have been employed to model interactions involving different entities. Coupled Hidden Markov Model (CHMM) are employed to model interactions between people (Oliver *et al.* , 2000), factoring the hidden Markov chains and distributions of the entities. The hidden Markov chains are coupled to make the state of an entity dependant on the state of the other entities at the previous instant.

TThe Markov assumption might be too rigid to model the temporal evolution of the hidden labels Y. To this end Hidden Semi-Markov Models (HSMM)(Hongeng

& Nevatia, 2003) have been developed, explicitly modelling the duration of hidden states instead of the exponential decay assumed by the Markov Assumption. The Semi-Markov assumption has been also introduced in CHMM models (Natarajan *et al.* , n.d.).

To model complex activity, with shared sub-events, HMMs can be defined hierarchycaly. Hierarchical Hidden Markov Models (HHMM) (Nguyen *et al.* , 1987) are built stacking multiple HMMs. The observations are introduced at the lowest level, and the inferred probability distribution of the hidden state sequence Y is fed as the observation for the next level. Efficient algorithms have been defined to compute the probabilities of the hidden states at different levels.

The usual criteria to train Hidden Markov Models is the maximization of the likelihood of the joint distribution of observations, given the sequence labels if available (in segmentation tasks) (Rabiner, 1989). Entropy minimization of the joint distribution (Brand & Kettnaker, 2000) has been shown as an alternative to obtain models with more compact parameters, allowing a better interpretation of the action dynamics and improving their predictive performance. Other alternative is to maximize the conditional likelihood of sequence labels given the observations (Pérez *et al.* , 2007), improving the predictive performance in sequence segmentation tasks.

When employing Hidden Markov Models the number of hidden states should be properly setup in order to prevent overfitting. There are different strategies to choose the appropriate number of hidden dimensions. Akaike Information Criterion and Bayesian Information Criterion (Xiang & Gong, 2006) are standard metrics employed to select the proper distribution. A specific scoring function for the hidden Markov model has been proposed (Xiang & Gong, 2008).

Other alternative is to employ Bayesian non-parametric modelling . The Infinite Hidden Markov Model employs a Dirichlet process prior to average the proper number of hidden states (Pruteanu-Malinici & Carin, 2008). It has shown good performance in abnormal sequence detection tasks.

Other family of generative models employed in motion analysis taks are the based on Linear Dynamical Systems (LDS). The discrete hidden variable of the HMM is replaced in the LDS by a continuous random vector, modeling the hidden parameters of the system under observation. Cascades of LDS have been employed for action sequence clustering (Turaga *et al.* , 2009). Non linear/non-parametric dynamical systems have been proposed under the gaussian process framewok(Wang *et al.* , 2008a). Distance metrics between sequences have been defined between their corresponding non-linear LDSs (Chaudhry *et al.* , 2009).

The main drawback of DBNs comes from their generative nature. Modelling the joint probability distribution $P(X, Y)$ requires from a large number of parameters. Learning algorithms to fit the joint conditional distribution parameters require a lot of training samples to obtain accurate estimates.

Discriminative models An alternative to the modelling of the joint probability distribution $P(X, Y)$ made by the generative models just introduced is to model the conditional probability distribution $P(Y \mid X)$ of the labels Y given the observations. Discriminative models have some theoretical advantages over generative methods: they directly model the conditional distribution that should be maximized in segmentation and classification tasks, requiring to adjust fewer parameters. Thus, the number of training samples required to learn accurate parameter estimations is smaller. By contrast, this models are not suitable for abnormality detection tasks.

The basic discriminative model for sequences is the Conditional Random Field (CRF), designed to perform sequence segmentation. The conditional distribution of a set of sequence labels $Y = (y_1 \dots y_T)$ given a sequence of observations $X = (x_1, \dots, x_T)$ is defined as $P(Y \mid X) = \frac{1}{Z} exp\left(\sum_t = 1^T \phi(y_t, y_{t-1}) + \phi(y_t, x_t)\right)$. The CRF might be seen as an structured extension of the logistic regression classifier·

The Hidden Conditional Random Field Model (HCRF) (Quattoni *et al.* , 2007) is an extension of the CRF to perform sequence classification. A set of hidden variables is employed to model sequence dynamics conditioned on the class label Y of the

sequence. The HCRF model has become very popular since its recent introduction, as it outperforms HMMs in sequence classification tasks. It has been extended with a root filter to take into account the compatibility of the observations and the sequence label (Wang & Mori, 2008a), improving the performance. The main drawback of the HCRF is that the training function employed is not convex by the presence of hidden variables. To overcome this limitation it has been formulated in a max-margin setting (Wang & Mori, 2008b), turning the optimization problem into a convex one at the cost of discarding the probabilistic modelling. Other alternative is to employ an auxiliary HMM to make the hidden variable correspondences visible (Zhang & Gong, 2010b),

The Latent-Dynamic Conditional Random Field (LDCRF) (Morency *et al.* , 2007)extends the CRF introducing hidden variables in a similar way to the HCRF, augmenting the predictive performance of CRFs in segmentation tasks.

Factorial Conditional Random Fields have been defined to model concurrent labels (Wu *et al.* , 2001). Label chains are coupled as in the FHMM. Dynamic Conditional Random Fields(Sutton *et al.* , 2007) generalize the possible factorizations made to CRFs as the DBNs generalize the factorizations made to HMMs. Semi-Markovian extensions of the CRF has been proposed also (van Kasteren *et al.* , 2010) , including high order interactions in the sequence dynamics and improving predictions accuracies. Hierarchical extensions of CRF (Liao *et al.* , 2007) have been introduced to model actions and places at the same time. Model and feature selection has been proposed for the CRF employing a l1 penalty (Vail *et al.* , 2007). Models trained in this way show a performance higher than those trained with the standard l2 penalty.

In fact the usage of discriminative action models is recent and not very widespread. However, their performance higher than the achieved by generative models in different tasks is popularizing them and novel extensions are proposed every year. This thesis contributes to their development providing model and feature selection algorithms for the HCRF.

2.5.1.3 Local feature action models

Bag of words models are related to the usage of local features. Each video sequence is represented by a set of spatio-temporal local descriptors computed from the detected salient points. The number of local descriptors extracted varies between sequences, complicating action modeling. Mechanisms to handle this peculiarity should be included in the action model, as the observation now is not a real valued vector.

Features are encoded as occurrences of a dictionary of visual words. The elementary approach to model actions employing local descriptors is to assume the exchangeability of the words, i.e., to discard their spatial and temporal ordering, and build models upon the frequency of occurrence of the words in the video. This models are known as "Bag of Words" and are inspired from the language recognition literature.

The basic bag of words model represent each video to classify with an histogram of visual word occurrences and introduces it into a classifier to predict the category of the action from a predefined set. The ξ^2 Support Vector Machine has been a typical choice of classifier in different works (Schuldt *et al.* , 2004).

Other methods model the joint probability distribution of the observed visual words and their corresponding action label. The simplest model is the Naive-Bayes classifier (Yang *et al.* , 2008), assuming that the observed features are conditionally independent given the action class. Probabilistic latent topic models such Probabilistic Latent Semantic Analysis (PLSA) (Wong *et al.* , 2007) and Latent Dirichlet Allocation (Niebles *et al.* , 2008) have shown a better performance, as they model the coocurrence of different words at the same time in the probability distribution.

The main problem of all these approaches is that they discard the spatio-temporal ordering of the detected features. A discriminative boosting framework including temporal ordering has been proposed to overcome this limitation (Nowozin *et al.* , 2007). PLSA has been extended to incorporate the probability of the detected features in a given order (Zhang & Gong, 2010a).

2.6 Remarks

This section has presented the most relevant works related to Human Action Recognition. It has been shown that to bridge the sematic gap between pixel intensity values and high level knowledge about the video content it is necessary to employ feature extraction techniques to describe the motions and to employ different models to encode the spatio-temporal correlations among the extracted features. The different meanings in the literature of words such action, motion or event have been reviewed and a new hierarchy to categorize the levels of motion has been proposed. Relevant feature extraction methods and action recognition models have been reviewed.

The main conclussion that might be achieved from this state of the art review presented here is that there has been a paradigm shift from the first action recognition models that had a clear sense of what they were modeling (arms, torso, legs) to recent approaches looking into the outputs of image filters without an *a priori* sense but capturing valuable information for the action recognition. The lack of interpretability of these models is compensated by the high accuracy shown. Actions are predicted in the presence of body limb occlusions that prevent recovering the whole body configuration. Invariance towards viewpoint changes has been achieved at the same time. Some proposals are already able to predict actions from images captured from viewpoints not known during system training.

The design of the lowest levels of action recognition systems has been widely explored. Future works will have to focus on the upper levels, as there is no general methods yet for activity and interaction recognition. Most of the current approaches are designed by hand. General learning methods to learn accurate rules to recognise these categories have to be explored in order to simplify the creation of the systems.

Beyond action modelling, understanding what is going to happen in the future might be another interesting line of research. This will lead to the identification of risky situation before they are produced. The soon they are identified, the soon they might be prevented.

All this facts let us think that the problem of human action recognition is not close from being solved and important contributions to the field have to be done in future years.

3

Data Fusion and Human Action Recognition

It was six men of Indostan
To learning much inclined,
Who went to see the Elephant
(Though all of them were blind),
That each by observation
Might satisfy his mind

The poems of John Godfrey Saxe

DATA fusion is the discipline studying the efficient combination of measurements obtained from multiple sensors in order to achieve more specific inferences than could be achieved by using a simple, independent sensor (Liggins *et al.* , 2008). Previous chapter has introduced the methods and techniques employed to perform human action recognition, with the focus on single camera systems. The purpose of this chapter is twofold: show how human action recognition relates to data fusion and employ data fusion concepts to provide a categorization of human action recognition systems employing multiple cameras.

The main attributes of data fusion systems are introduced in first term. The JDL data fusion process model and Dasarathy's input-output model are presented as different frameworks to categorize data fusion systems. This frameworks are then em-

ployed to analyze human action recognition systems, with the focus on multiple camera human action recognition. The categorization presented here will be employed to locate some of the contributions of this dissertation presented in subsequent chapters.

3.1 Data fusion

Data Fusion studies the efficient combination of measurements obtained from multiple sensors or, alternatively, the temporal measurements obtained from a single sensor, in order to achieve more specific inferences about the state of one or more entities than the ones that could be achieved by using a single, independent, sensor (Liggins *et al.* , 2008).

Although the data fusion concept is not new - human brain is a data fusion system that efficiently combines the data gathered by the five senses-, the concepts, methods and computational models for data fusion have been developed from the 80's, after the great interest shown in the area by the defense community. The interest is reflected in the amount of research and industrial contracts funded by the US Department of Defense to develop new data fusion algorithms and applications. This large funding, together with the improvements made to sensor technologies, computational architectures and communication networks have enabled the creation of real-time data fusion applications unbelievable some decades ago. But the usage of data fusion concepts and architectures is not limited to the defense domain. Different civil applications, involving the processing of data gathered by multiple sensors, have been benefited from the advances made in the area. Robotics, Aerospace, Medical Systems or Ambient Intelligence have successfully applied Data Fusion concepts to improve their developments.

Data Fusion is an heterogeneous discipline, bringing the theoretical developments of multiple disciplines to practice. The theory of signal processing, artificial intelligence, computer networks, control theory or statistical estimation, among

others, is combined to solve the problem of inferring the state of some entities of interest.

The basis of Data Fusion is the usage of multiple measurements sampled at different spatial or temporal locations, instead of a single measure at a given spatio-temporal location. When identical sensors are employed from the same location, combining their measurements in the proper way leads to a more accurate state estimation, washing noise artifacts and preventing the effects of sensor malfunction. When identical sensors are placed at different locations, complementary measurements of the target are obtained, and the optimal combination of them, under appropriate constraints, leads to an state estimation better than the sum of the parts. An example of this behavior is 3D visual hull reconstruction from multiple 2D silhouettes. The 3d visual hull has more information than the silhouettes by themselves, caused by the alignment of the data in the 3D space. When the field of view of the sensors is not fully overlapped, wider areas can be observed, extending the coverage achieved by a single sensor. Last, but not least, when employing different kind of sensors, the fusion of their measurements leads to a better estimate of the target state. An example of this is the Microsoft Kinect device, combining depth and color measurements to provide complimentary estimations of the player appearance.

The formal definition of Data Fusion was given by the Joint Directors of Laboratories Data Fusion Working Group in 1985, as *A process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates, and complete and timely assessments of situations and threats, and their significance. The process is characterized by continuous refinements of estimates, assessments and the evaluation for the need of additional sources, or modification of the process itself, to achieve improved results* (White *et al.* , 1988). In fact, this definition has shown to be too restrictive with the years, as it is focused on defense applications. Similar underlying problems of data association and combination occur in a very wide range of engineering, analysis and cognitive situations and are not covered by the definition. By

this reason, among others, the formal definition of data fusion was simplified in 1998 to *The process of combining data or information to estimate or predict entity states* (Steinberg *et al.* , 1998). This simplification makes the scope of data fusion wider, as the estimation of position and identity is replaced by the more generic "entity states". Now this estimations are not required to be complete and timed, covering a broader range of possible inference techniques.The removal of association is motivated because it is not something required in every data fusion application, while the removal of correlation comes from the fact that is just an statistical technique that might or might not be used.

But Data Fusion is not the solution to every problem, as it has some well defined limitations (Liggins *et al.* , 2008):

- *There is no substitute for a good sensor*. If the state to be inferred does not produce effects observable by the employed sensor, it does not matter how much effort is put into the data fusion process. No accurate state estimations will be obtained.

- *Downstream processing cannot absolve the sins of upstream processing*. The best possible processing should be made at each level/step. Failure in the design of the lower levels unnecessary complicates the processing at the upper levels, without ever obtaining the performance that would lead the right processing at lower levels.

- *The fused answer may be worse than the best sensor*. When the quality of each sensor measurement is not properly estimated, a high importance can be given to untrusted sources, corrupting the result of the fusion as good measurements get corrupted.

- *There are no magic algorithms*. The algorithm with optimal performance for every situation does not exist. Different situations require from different techniques with different operating characteristics. There is no free lunch.

- *There will never be enough training data*. Pattern recognition methods employed for data fusion are estimated from training samples. However, there is usually a bias between the samples used during training and the samples used during operation. The bias decreases as the number of training samples grows, but it is almost impossible to reflect all the possible operation conditions during training. This fact reduces the performance of data fusion systems.

3.2 Characterization of Data fusion systems

This section presents the JDL process model and Dasarathy's input-output model. These are complementary frameworks for the analysis of data fusion systems whose usage is widely extended.

3.2.1 The JDL Process Model

The JDL Data Fusion model (White *et al.* , 1988) is the most widely used framework for the categorization of data fusion systems and algorithms. The first version was published in 1985 by the US Joint Directors of Laboratories (JDL) Data Fusion Working Group with the aim of providing a common framework to facilitate the communication between the communication between data fusion stakeholders and provide a conceptual framework for new developments. The JDL model is not an architectural paradigm nor a process model for the creation of data fusion system. Instead, it provides different levels of abstraction where the different algorithms employed in data fusion systems might be accommodated according to the kind of processing they perform.

The stated purpose for that model and its subsequent revisions have been to:

- Categorize different types of fusion processes.

- Provide a technical architecture to facilitate reuse and affordability of data fusion and resource management system development.
- Provide a common frame of reference for fusion discussions.
- Facilitate understanding of the types of problems for which data fusion is applicable.
- Codify the commonality among problems.
- Aid in the extension of previous solutions.
- Provide a framework for investment in automation.

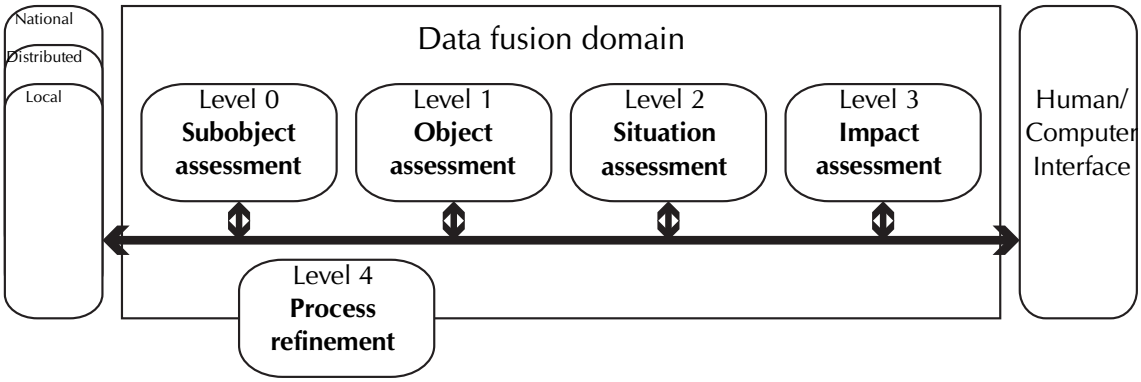


Figure 3.1: The JDL data fusion model (1998 revision)

The JDL data fusion model, after the 1998 revision (Steinberg *et al.* , 1998), proposes five different levels of abstraction where the data fusion functions are accommodated (figure 3.1. These levels are:

- Level 0. *Signal/Feature Assessment*. This level includes the algorithms employed to enhance or combine the input signals of the fusion systems. The inferences made at this level do not make any assumption about the causes originating the signals. Typical operations at this level include spatial and temporal data alignment, data standardization and data preconditioning for bias removal.

- Level 1. *Entity Assessment*. Algorithms employed for the estimation of the current state of a individual entities are defined at this level. This includes target detection, classification, location, tracking and identity estimation. Processing at this level usually implies the association of observations to the corresponding responsible targets.

- Level 2. *Situation Assessment*. A situation is *a set of entities, their attributes, and relationships*. Thus, the task at this processing level is to infer the existent relationships between the analyzed entities employing the individual state estimations.

- Level 3. *Impact Assessment*. The purpose of the algorithms defined at this level is to predict future situations derived from the current and past inferred situations. This includes the computation of expected outcomes of actions executed to alter the current situation or the projection of the current situation to the future to predict the possible evolution.

- Level 4. *Process Assessment*. This level includes the algorithms employed to measure the real-time performance of the fusion system and improve it. This includes the reconfiguration of the sensors employed or the replacement of data fusion algorithms by others better adapted to the current or expected scenario.

3.2.2 Dasarathy's Input-Output Model

Dasarathy proposed an alternative categorization of Data Fusion systems according to the level of abstraction of the information at the input and output of the fusion system (Dasarathy, 1997). Three different levels of abstraction are defined: (1) *data*; (2) *features* and (3) *decisions*. Data is the lowest level of abstraction, corresponding to the raw measurements of the sensors, such pixel intensities or depth information. Features are transformations of the data to enhance some property such edges or curvature. Finally, decisions encode information about the certainty of a fact, in the

form, among others, of probability estimates or fuzzy sets.

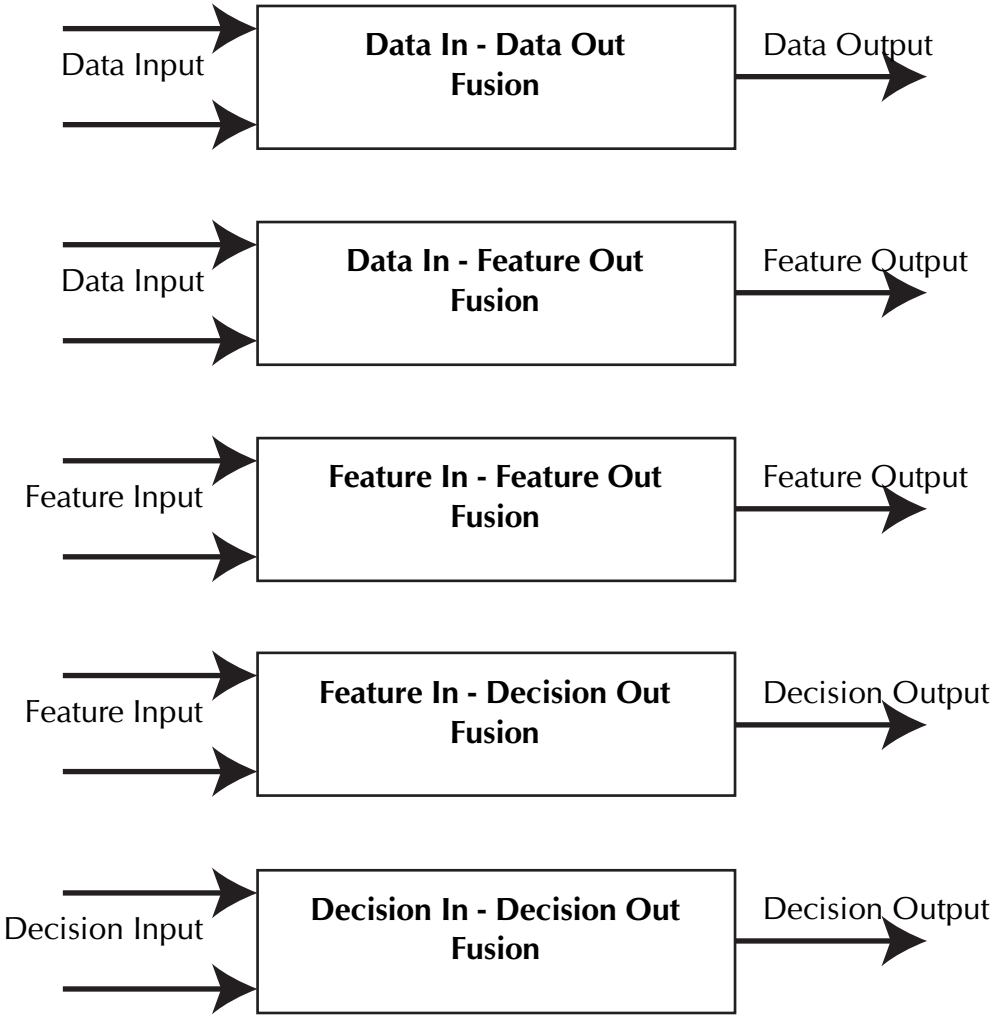


Figure 3.2: Dasarathy Input-Output model

Data fusion systems are characterized according to this abstraction of their inputs and outputs as follows (figure 3.2):

- Data in-Data out (DAI-DAO) Fusion. At the lowest level of abstraction are systems processing *data* and generating *data*. An example of this kind of fusion systems are multispectral imaging devices: pixel intensities are captured at different wavelengths to compose an image better describing the reality. High Dynamic Range (HDR) imaging is another example of a DAI-DAO fusion system, combining images taken with different exposition configurations to have

a better representation of the details of dark and light regions of the scene.

- Data in- Feature Out (DAI-FEO) Fusion. At the next level of abstraction in the hierarchy are the systems processing *data* to generate *features*. Stereo vision systems are located at this level, as they compute disparity maps (*features*) from pixel intensities (*data*).

- Feature in-Feature Out (FEI-FEO) Fusion. At the mid level of the hierarchy are located systems processing *features*. The conceptually simpler are those generating *features* too. Due to the vague definition of what is a feature at this category lie a wide variety of systems. Fusion systems combining the measurements of the same state variable to provide a more robust estimation of the real value belong to this category.

- Feature In-Decision Out (FEI-DEO) fusion. The next abstraction level is related to pattern recognition systems, transforming *features* into *decisions* about the class of the phenomena being recognized. At this level are defined those data fusion systems based on introducing a set of features computed from multiple sources into a classifier.

- Decision In-Decision Out (DEI-DEO) fusion. The highest level of abstraction includes the system that combine independent decisions about the phenomena to study to make a global decision about it. Decisions might be defined in different forms, such crisp values, probabilistic distributions or fuzzy sets.

3.3 Human Action Recognition from the data fusion perspective

Once that the main concepts of data fusion systems have been presented, Human action recognition is going to be analyzed employing them. First, Human Action Recognition is characterized under the JDL process model in general terms. Then,

the Dasarathy input-output model is employed for the characterization of works in Human Action Recognition from multiple cameras.

Human Action Recognition applications may be considered from the viewpoint of data fusion, as they process signals from - possibly - multiple sensors to obtain high level knowledge of the motion being performed by the observed human.

As already seen in previous chapter, temporal integration is a key element of most Human Action Recognition systems, as actions are performed along time. However, Human Action Recognition applications have not been studied using the concepts of the data fusion community, as it has been an application traditionally studied from the pattern recognition domain. The discussion in this chapter wants to relate Human Action Recognition with data fusion. To this end, the data fusion models are presented and then related to Human Action Recognition techniques.

3.3.1 Human Action Recognition and the JDL process model

The Human Action Recognition algorithms introduced in previous chapter are defined at the level 1 of the JDL process model, as they are related to the assessment of the state of individual entities. The state variable to infer is a label characterizing the kind of action. From the definition of the JDL level 0 there might be a temptation of defining the feature extraction algorithms at that level. However, level 0 is related to functions not considering individual entities, and most of the presented features are computed from segmented humans. Local features are the exception, as they not require a previous segmentation, but as their existence is related to the recognition of actions and do not have any meaning by themselves, so they should be located also at JDL level 1.

The recognition of interaction tasks as defined on previous chapter is located at JDL level 2. This includes the interactions between humans or humans and objects.

Level 3 in Human Action Recognition corresponds to the prediction of the future actions that person is going to do. However, to the best of our knowledge no applica-

tions at this level have been defined. The plan recognition problem (Kautz & Allen, 1986), where the objective is to infer what is goal of an observed agent would be the closer sample to this level.

Levels 4 and 5 of the JDL process models have not been very exploited from human the human action recognition perspective. Level 4 studies how the information is presented to the system operator. Commercial video surveillance applications incorporate this capabilities, incorporating semantic information in the reports. Commercial gaming platforms with visual inputs represent the motions performed by the player with avatars. Fitness trainers represent with them how the player is performing a given exercise and how they should do it, in order to correct their performance and prevent hurts.

Level 5 would study the adaption of the algorithms employed to new conditions of the environment, such lighting or occlusions. However, to the best of our knowledge, no works have been reported proposing such applications.

3.3.2 Human Action Recognition and Dasarathy's input output model

The Dasarathy's input-output model provides a framework to categorize the works in Human Action Recognition employing multiple views of the scene being analyzed. From the viewpoint of this dissertation is the most interesting one, as it provides a framework to easily categorize multiple camera human action recognition applications.

The employment of multiple views in human action recognition has some advantages over traditional single view approaches. Among others, the most important are:

- Viewpoint invariance. The appearance of actions changes according to the orientation in the execution in the action with respect to the camera. Thus, employing multiple views provides complementary information to achieve a more robust recognition.

- Robustness towards occlusions. In real environments there is usually multiple furniture, walls or other objects that produce partial occlusions in the observed target. The way to overcome this limitation and not loss important motion information is to observe the scene from multiple viewpoints.

- Wider scene coverage. A single camera has a very limited coverage. Multiple cameras are needed to cover full scenes.

Human Action Recognition methods employing multiple cameras are defined at FEI-FEO, FEI-DEO and DEI-DEO levels. Although fusion at the data levels might be employed for human action recognition, they are not considered, as this kind of fusion is independent of the higher level task.

Diverse methods have been defined at the FEI-FEO data fusion level to combine the information obtained from multiple cameras. Different strategies have been defined at this level. It is possible to divide this works in three different categories: (1) methods projecting 2D features to 3D; (2) methods combining features in a subspace; (3) methods selecting the best available view.

Different 3D representation might be obtained from projecting 2D features to 3D. A popular approach is to recover the 3D shape projecting 2D silhouettes and recovering the visual hull(Gkalelis *et al.* , 2009; Pehlivan & Duygulu, 2011; Peng *et al.* , 2009). Visual hull reconstruction requires accurate silhouette segmentation at the different available views. Recent works have proposed alternatives based on the projection of optical flow to 3D (Holte & Chakraborty, 2011), or the projection of local interest points (Holte *et al.* , 2011b). Other works recover the 3D star skeleton by the correspondence of the corresponding 2D skeletons (Chen *et al.* , 2008). The correspondence between action sketches might be computed from multiple views (Yan *et al.* , 2008). The main drawback of all these approaches is that they need from accurate camera calibration parameters to perform the projection of the features in 3D.

Alternative methods compute features for the 2D views available and combine

them employing some simple scheme. The averaging of the multiple features representing pose, global and local motion has been proposed improving the results with respect to other alternatives (Määttä & Aghajan, 2010). A joint Bag-of-Words histogram might be constructed with the local feature descriptors obtained for each one of the views (Wu *et al.* , 2010), but a higher performance is obtained with other fusion strategies. Projections maximizing the cross-covariance between the $\mathcal{R}$-transform derivatives computed at each view have been defined to learn a joint subspace where the action recognition is performed (Karthikeyan *et al.* , 2011). Two level Linear Discriminant Analysis is employed to learn silhouette projections maximizing the separability of the action classes (Iosifidis *et al.* , 2012). All this methods provide more flexible solutions for the combination of the features obtained from multiple cameras. However, the experimental results show a lower performance than the methods based on 3D reconstruction.

The last class of methods is based on computing a measurement of the quality of each view available, in order to select the best and perform the recognition with the data from that view. A first approach to the selection of the best view is made estimating the orientation of the human with respect to the camera (Shen *et al.* , 2007). A measurement based on the properties of the silhouette has been proposed (Määttä & Aghajan, 2010). Other proposed measure in the case of employing local features is to choose the camera with the highest number of detections (Wu *et al.* , 2010). Different utility measures have been proposed studying the saliency, concavity or variations of silhouette stacks (Rudoy & Zelnik-Manor, 2011). The main drawback of this approaches is that they do not exploit the complementary information that might be present at each view.

The next category of works examined employing multiple views of the scene for the recognition of human actions are those defined at the FEI-DEO level. This works model the existing correlations among the multiple observations in the structure of the classifier employed for the prediction of the actions. The concatenation of the input features is the most straightforward procedure to perform the fusion (Määttä &

Aghajan, 2010; Wu *et al.* , 2010). The Fused HMM (Wang *et al.* , 2007b) proposes to model correlations among observations coupling the values of the hidden state chains of parallel HMMs defined for each view. Histograms of local features have been fused rotating the ordering of the inputs to account for the variations in the orientation of the inputs (Srivastava *et al.* , 2009). The main drawback of this works is their lack of flexibility, assuming that the camera configurations remain unchanged between train and test steps. A procedure for the alignment of camera views where the configuration changes from train to test steps is defined in (Ramagiri *et al.* , 2011), but requiring the knowledge of relative camera placement.

The last category of works employing multiple views performs the fusion at the DEI-DEO level, combining the outputs of action classifiers applied to each one of the camera views. Majority voting has been the most common technique for the fusion of decisions (Määttä & Aghajan, 2010; Naiel & Abdelwahab, 2010). A weighted voting strategy has been proposed in (Zhu *et al.* , 2012), correcting each vote according to the value of the observed feature.

3.4 Remarks

This section has presented the concepts and frameworks employed by the data fusion community and related them to human action recognition. The different levels of the JDL process model have been compared to the different steps needed to perform human action recognition. It has been shown that most of the human action recognition algorithms are defined at JDL level 1. At level 2 are defined algorithms studying interactions. Other levels have not been really exploited and they should be targets of future research.

Dasarathy's Input-Ouput hierarchy has been employed to categorize multicamera human action recognition applications. Existing works have been categorized under three conceptual classes according to the data abstractions employed. This categorization will serve us to classify the algorithms for human action recognition from

multiple cameras that are going to be introduced in subsequent chapters.

II

Proposals

4

Model and Feature Selection in Hidden Conditional Random Fields

Entia non sunt multiplicanda praeter necessitatem

William of Ockham

THE importance of sequence modeling methods in Human Action Recognition has already been pointed in previous chapters. The Hidden Conditional Random Field (HCRF) is a discriminative model employed in sequence classification tasks that has shown a high predictive performance in experimental evaluations, outperforming other existing methods. However, the standard algorithm to estimate the optimal parameter values for the HCRF from a set of training samples does not incorporate model and feature selection capabilities. HCRFs trained with this method are too complex, modeling noisy attributes of the training samples that in fact do not provide any information to the classification process. The definition of a training procedure reducing the complexity of the result HCRF by model and feature selection will lead to an increase of the experimental predictive performance.

This chapter presents special training algorithms performing model and feature selection. The HCRF model with the standard training procedure is introduced in first term, pointing out their limitations. Then the proposed training procedure is presented. Experimental evidence in given to show that the proposed method has a higher predictive performance than the standard training method.

4.1 Hidden Conditional Random Fields

The HCRF (Quattoni *et al.* , 2007) is an undirected graphical model that belongs to the exponential family. It might be understood as an extension of the Conditional Random Field incorporating hidden variables to model the correlations among the different observations. Different structured prediction tasks might be tackled with HCRFs, but this work assumes without lot of generality sequence classification.

Formally, the HCRF defines the conditional probability distribution of a discrete random variable $y \in \{y_1, \dots, y_N\}$ (a.k.a. sequence label) given a sequence of random variables $\mathbf{x} = x_1, \dots, x_T$ (a.k.a. observations) employing a set of auxiliary discrete hidden variables $\mathbf{h} = h_1, \dots, h_T$, $h_i \in \mathcal{H}$ not observed during training. These variables are introduced to model correlations among the observations in $\mathbf{x}$. In the case of sequence classification, these correlations correspond to the sequence dynamics. The conditional probability of the sequence label y and the hidden variable assignments $\mathbf{h}$ given the sequence of observations $\mathbf{x}$ is defined using the Hammersley-Clifford theorem of Markov Random Fields:

$$P(y, \mathbf{h} \mid \mathbf{x}, \theta) = \frac{e^{\Psi(y, \mathbf{h}, \mathbf{x}; \theta)}}{\sum_{y'} \sum_{h} e^{\Psi(y, \mathbf{h}, \mathbf{x}; \theta)}} \tag{4.1}$$

The conditional probability of the class label y given the observation sequence $\mathbf{x}$ is obtained marginalizing over all the possible value assignments to hidden parts $\mathbf{h}$:

$$P(y \mid \mathbf{x}, \theta) = \frac{\sum_{h} e^{\Psi(y, \mathbf{h}, \mathbf{x}; \theta)}}{\sum_{y'} \sum_{h} e^{\Psi(y', \mathbf{h}, \mathbf{x}; \theta)}} \tag{4.2}$$

The potential function $\Psi(y, \mathbf{h}, \mathbf{x}; \theta)$ measures the compatibility of the input $\mathbf{x}$ with the assignments to the hidden variables $\mathbf{h}$ and the class label y. There are multiple

posibilities about the form of this function. Here it is defined as:

$$\Psi\left(y, \mathbf{h}, \mathbf{x}; \theta\right) = \sum_{t=1}^{T} \phi\left(x_t\right) \alpha(h_t) + \sum_{t=1}^{T} \beta(h_t, y) + \sum_{t=1}^{T} \gamma\left(h_t, h_{t+1}, y\right) \tag{4.3}$$

where $\phi\left(x_t\right) \in \mathcal{R}^d$ is the feature vector associated with the observation x_t and $\theta = [\alpha \quad \beta \quad \gamma]$ is the vector of model parameters, indexed according to the values given to the hidden variables $\mathbf{h}$ and label y. The first term, parameterized by $\alpha\left(h_t\right) \in \mathcal{R}^d$ measures the compatibility of the observation at instant x_t with the assignment to the hidden variable h_t. The second term measures the compatibility of the values given to the hidden parts h_t with the class label y and is parameterized by $\beta\left(y, h_i\right) \in \mathcal{R}$. Finally, the third term, parameterized by $\gamma\left(y, h_t, h_{t+1}\right) \in \mathcal{R}$ models sequence dynamics, measuring the compatibility of adjacent hidden variable assignments h_t and h_{t+1} with the class y.

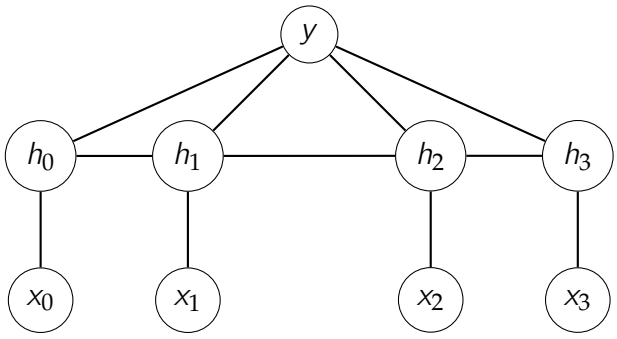


Figure 4.1: Graphical model representing the structure of the HCRF induced by the function Ψ

The function Ψ induces the structure of the undirected graphical model defined by the HCRF. The structure of this graph can be observed on figure 4.1. Exact inference of the conditional probability distribution defined in equation 4.2 is possible, as the dependencies among the values given to the hidden variables $\mathbf{h}$ form a chain. Efficient inference is achieved employing belief propagation (Bishop *et al.* , 2006).

4.1.1 Parameter estimation

Optimal model parameters θ^* are estimated from a set of K training samples $\left(\mathbf{x}^i, y^i\right), 1 \leq i \leq K$, minimizing the L_2 regularized negative conditional log-likelihood function:

$$\theta^* = \arg\min_{\theta} L(\theta) = \arg\min_{\theta} -\sum_{i=1}^{K} \mathcal{L}\left(\mathbf{x}^i, y^i; \theta\right) + \lambda R(\theta). \tag{4.4}$$

The first term measures how model parameters are adjusted to predict each one of the K training samples, while the second term acts as a regularization prior over model parameters. The standard regularization employed in the HCRF is the Ridge regularizer, defined as $R(\theta) = ||\theta||_2^2$, imposing a zero-mean gaussian prior on the values of θ to prevent overfitting. The parameter λ defines a tradeoff between regularization and adjustement. A value of $\lambda = \frac{1}{2\sigma^2}$ is equivalent to a gaussian with variance σ^2. The conditional log-likelihood function $\mathcal{L}(\mathbf{x}, y; \theta)$ is defined as:

$$\mathcal{L}(\mathbf{x}, y; \theta) = \log P(y \mid \mathbf{x}, \theta) = \log\left(\frac{\sum_h e^{\Psi(y, \mathbf{h}, \mathbf{x}; \theta)}}{\sum_{y'} \sum_h e^{\Psi(y', \mathbf{h}, \mathbf{x}; \theta)}}\right) \tag{4.5}$$

Due to the presence of the hidden variables $\mathbf{h}$, the objective function in equation 4.4 is non-convex (Boyd & Vandenberghe, 2004). However, a local optimum θ^* for the model parameter values might be obtained employing standard convex optimization techniques, as the function in 4.4 has an smooth gradient. The partial derivative of $\mathcal{L}(\mathbf{x}, y; \theta)$ with respect to each component $\theta(h_i)$ parameter is given by:

$$\begin{aligned}
&\frac{\partial \mathcal{L}(\mathbf{x}, y; \theta)}{\partial \theta(h_i)} = \\
&= \sum_{\mathbf{h}} P(\mathbf{h} \mid y, \mathbf{x}; \theta) \frac{\partial \Psi(y, \mathbf{h}, \mathbf{x}; \theta)}{\partial \theta(h_i)} - \sum_{\mathbf{h}, y} P(y, \mathbf{h} \mid \mathbf{x}; \theta) \frac{\partial \Psi(y, \mathbf{h}, \mathbf{x}; \theta)}{\partial \theta(h_i)} \\
&= \sum_{j,a} P(h_j = a \mid y, \mathbf{x}; \theta) x_j - \sum_{j,a,y} P(y, h_j = a \mid \mathbf{x}; \theta) x_j
\end{aligned} \tag{4.6}$$

Similarly, partial derivatives w.r.t $\theta(h_t, y)$ parameters are given by:

$$\begin{aligned}
&\frac{\partial \mathcal{L}(\mathbf{x}, y; \theta)}{\partial \theta(h_i, y)} = \\
&= \sum_{\mathbf{h}} P(\mathbf{h} \mid y, \mathbf{x}; \theta) \frac{\partial \Psi(y, \mathbf{h}, \mathbf{x}; \theta)}{\partial \theta(h_i, y)} - \sum_{\mathbf{h}, y} P(y, \mathbf{h} \mid \mathbf{x}; \theta) \frac{\partial \Psi(y, \mathbf{h}, \mathbf{x}; \theta)}{\partial \theta(h_i, y)} \\
&= \sum_{j,a} P(h_j = a \mid y, \mathbf{x}; \theta) - \sum_{j,a,y} P(y, h_j = a \mid \mathbf{x}; \theta)
\end{aligned} \tag{4.7}$$

Finally, partial derivatives w.r.t. $\theta(h_t, h_{t+1}, y)$ parameters are given by:

$$\begin{aligned}
&\frac{\partial \mathcal{L}(\mathbf{x}, y; \theta)}{\partial \theta(h_t, h_{t+1}, y)} = \\
&= \sum_{\mathbf{h}} P(\mathbf{h} \mid y, \mathbf{x}; \theta) \frac{\partial \Psi(y, \mathbf{h}, \mathbf{x}; \theta)}{\partial \theta(h_t, h_{t+1}, y)} - \sum_{\mathbf{h}, y} P(y, \mathbf{h} \mid \mathbf{x}; \theta) \frac{\partial \Psi(y, \mathbf{h}, \mathbf{x}; \theta)}{\partial \theta(h_t, h_{t+1}, y)} \\
&= \sum_{t,a,b} P(h_t = a, h_{t+1} = b \mid y, \mathbf{x}; \theta) - \sum_{t,a,b,y} P(y, h_t = a, h_{t+1} = b \mid \mathbf{x}; \theta)
\end{aligned} \tag{4.8}$$

The conditional probabilities appearing on equations 4.6,4.7,4.8 are efficiently estimated employing belief propagation.

Different search strategies might be employed to find the optimal parameter values. might be applied employing the gradient just introduced to find the minima of the objective function. Among them, the LBFGS quasi-newton method is the most popular (Zhu *et al.* , 1997), updating the descent direction with an approximation of the Hessian based on previous gradient estimations. Others have proposed to employ an online stochastic gradient descent algorithm (Zhu *et al.* , 1997), achieving a fast convergence rate but at the cost of obtaining a worst quality solution. Stochastic gradient descent is indicated for large scale learning scenarios with a huge number of training sequences. In any case, the non-convexity of the objective function to optimize makes necessary to run the search multiple times from different starting points.

4.1.2 Limitations

The standard method to estimate HCRF optimal parameters leaves some open issues that are going to be discussed in order to motivate the proposal in subsequent section. These are:

- **How to adjust the dimensionality of hidden state variables?**. $|\mathcal{H}|$ i.e., the number of different values that the hidden state variables in $\mathbf{h}$ can take, should be specified before parameter estimation. If very few values are given, the model will not have enough expressivity to capture all the correlations needed to predict class values. However, if too many values are given, noisy correlations are modeled, reducing the predictive performance of the obtained model. Thus, it is necessary to select the proper number of values. In practice this is done employing cross-validation, evaluating the predictive performance of optimal parameters for different choices to select the best. The non-concavity of the loss function in equation 4.4 makes the problem even worse, as many trials should be made per choice in order to obtain a good estimation of the optimallity of each configuration. Thus, an efficient procedure to estimate the right number of hidden variable qvalues is needed.

- **What if there are irrelevant variables in the input feature vectors $\phi(x_t)$?** The nature of the gradient of the L2 norm in equation 4.4 gives a non-zero weight to the parameters $\alpha(h_t)$ corresponding to irrelevant features. This fact reduces the predictive performance of the trained model, as irrelevant features in the input are taken into account when predicting the class of new samples. Thus, it is necessary to incorporate a method to select relevant features in the input and discard the irrelevants.

Other problem in the estimation of optimal HCRF parameters is how to adjust the tradeoff between parameter fitting and regularization, i.e., what value give to λ in equation 4.4. This problem is shared by every log-linear model trained with

regularization. In practice, λ is adjusted employing cross-validation, needing to try different values until the one with the best results is obtained. This adds another cross-validation dimension, as it should be already employed in the selection of the right number of hidden state values. The problem of estimating the right value for λ is out of the scope of this dissertation.

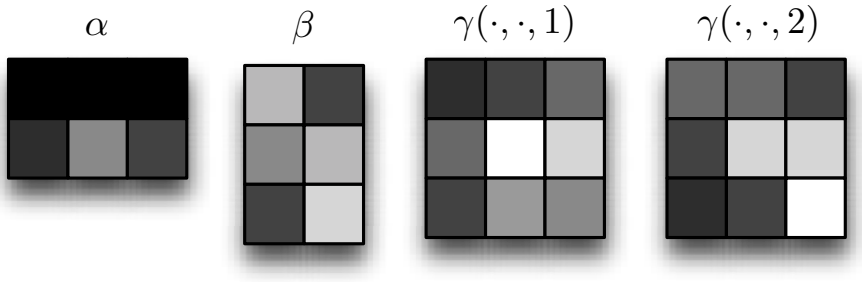
4.2 Model and Feature Selection in Hidden Conditional Random Fields

This section presents an overlapping group-L1 regularization strategy to estimate optimal parameters for the HCRF sequence classifier presented in previous section.

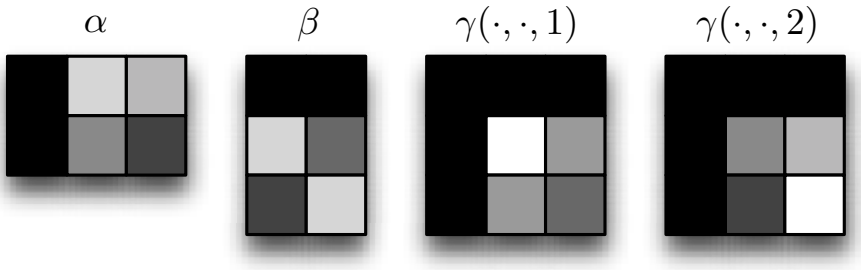
As described in previous section the components of the HCRF parameter vector θ are divided into three groups $\alpha(h_t)$, $\beta(h_t, y)$ and $\gamma(h_t, h_{t+1}, y)$, respectively indexed by the values of h_t, h_t and y and h_t, h_{t+1} and y. To obtain a model selection effect it is necessary to obtain a zero value in all the parameters related to each possible value of an unnecessary h. In a similar way, to obtain a feature selection effect, it is necessary to get a zero value for all the parameters related to an unnecessary input feature. The kind of target models that want to be obtained are shown on figure 4.2. It shows an HCRF with $d = 2$, $|H| = 3$ and $|Y| = 2$. Figure 4.2a shows that the parameters belonging to the first observation feature have got a zero value (dark colour). Similarly, figure 4.2b shows that the parameters for the first hidden variable have got a zero value. Both effects at the same time are presented in figure 4.2c.

Model and feature selection in log-linear models have been achieved replacing the L2 regularization term by a L1-regularizer (Ng, 2004), whose gradient leads to sparse solutions. However, L1-regularization is insufficient to achieve the desired effect, as it only gurantees to get zero values on single varibles and not on groups of them.

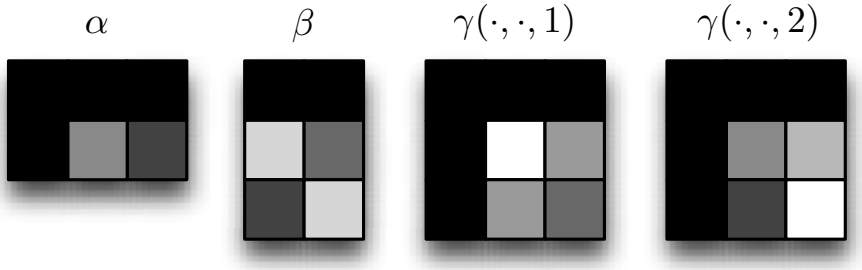
The solution to the problem is given by the usage of an overlapping group L1



(a) Feature Selection



(b) Model Selection



(c) Model and Feature Selection

Figure 4.2: The parameters of an small HCRF after feature selection, model selection and model and feature selection

regularization strategy (Huang & Zhang, 2010; Szabó *et al.* , 2011). Be $\mathcal{G}$ the power set of the parameter vector θ, and $G \subseteq \mathcal{G}$ a subset of the power set. The overlapping group-L1 regularized training of the HCRF is defined by:

$$\theta^* = \arg\min_{\theta} \mathcal{L}(\theta) + \sum_{g \in G} \lambda_g \left\|\theta_g\right\|^2 \tag{4.9}$$

The overlapping group-L1 norm sums the L2 norm of the different groups defined. At the optimal, some of the groups will have a zero norm, because all the parameters on that groups will have become zero. Depending on the way G is defined, model

selection, feature selection, both or even other advanced selection effects might be achieved:

- If $G \equiv G_{fs} = \cup_{d=1}^{D} \{\alpha(\cdot)_d\}$ feature selection is performed, as the L2 norm of the input features is penalized. A zero weight is expected for all the parameters corresponding to an input feature. Note that beta and gamma parameters are also regularized in order to prevent a big value on them, causing overfitting.

- If $G \equiv G_{ms} = \cup_{h=1}^{|\mathcal{H}|} \{\alpha(h) \cup \beta(h,\cdot) \cup \gamma(h,\cdot,\cdot) \cup \gamma(\cdot,h,\cdot)\}$ model selection is performed, as the L2 norm of the parameters corresponding to a hidden variable is minimized. A zero weight is expected to the parameters corresponding to non necessary hidden parts.

- If $G \equiv G_{fs} \cup G_{ms}$ both model selection and feature selection are performed at the same time.

4.2.1 Optimization algorithms

The convex optmization methods employed in the estimation of the optimal model parameters of the standard L2-regularized HCRF are no longer valid to recover the optimal parameter values of the overlapping group-L1 regularized objective function formulated on equation 4.9. The new regularization term makes the objective function non-smooth. In particular, the gradient has a singularity at the points where a group has a zero L2 norm. It is necessary to transform the problem into a smooth one in order to apply a gradient based method.

The unconstrained optimization problem in equation 4.9 might be reformulated

into an equivalent constrained optimization problem as suggested by (Schmidt, 2010):

$$\begin{aligned} \theta^* &= \min_{\theta} \mathcal{L}(\theta) + \sum_{g \in G} \lambda_g h_g \\ s.t. & \\ \forall g \quad & \|\theta_g\|_2 \leq h_g \end{aligned} \tag{4.10}$$

The overlapping group-L1 regularization term has been replaced by a set of constraints, one for each group of variables in G. Each one of the constraints in the optimization problem above defines a norm cone of radius h_g that ensures that the L2 norm of each group is smaller than h_g. A norm cone is a convex set, and the intersection of a set of convex sets is also a convex set (Boyd & Vandenberghe, 2004). Thus, the feasible region defined by the restrictions is convex. The norms of the different groups are added to the objective function. At the optimum the constraints are fulfilled with equality (it is trivial to probe that if they are not then it is not the optimal).

The objective function of the optimization problem in equation 4.10 is smooth, as the cause for the singularities has been removed. The estimation of the optimal parameters can be made employing a gradient descent method, projecting the obtained values into the feasible set defined by the restrictions.

Dykstra's algorithm (Bauschke & Lewis, 2000) solves the problem of projecting a point $w_0 \in \mathcal{R}^k$ into the intersection of a set of convex sets $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_q$, alternately projecting the point into each set and removing the residual from the previous step. The pseudocode for Dykstra's algorithm might be observed on listing 1

The projection operator $\mathcal{P}_{C_i}$ is the solution to the projection of a point into the set C_i, in this case the norm cone of size h_g. The projection is obtained solving the

Algorithm 1 Dykstra's cyclic projection algorithm

$\forall_i, l_i \leftarrow 0$;
$j \leftarrow 0$;
while w_j is changing by more than ϵ **do**
  for $i = 1$ to q **do**
    $w_j \leftarrow \mathcal{P}_{C_i}(w_{j-1} - l_i)$
    $l_i \leftarrow w_j - (w_{j-1} - l_i)$;
    $j \leftarrow j + 1$;
  end for
end while

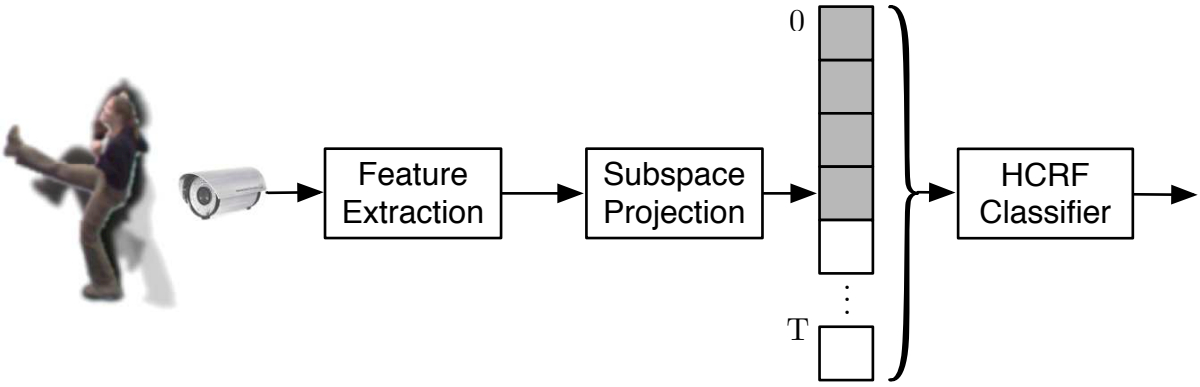
optimization problem:

$$\begin{aligned} &\min_{w' \in \mathbb{R}^n} \|w' - w\| \\ s.t. \quad & \\ &\|w'\| \leq g \end{aligned} \tag{4.11}$$

The solution to this optimization problem is given by the following equation, as shown on (Boyd & Vandenberghe, 2004):

$$\mathcal{P}_{\mathcal{C}_2}(x, g) = \begin{cases} (x, g) & \text{if } \|x\|_2 \leq g \\ \left(\frac{x}{\|x\|_2}\frac{\|x\|_2+g}{2}, \frac{\|x\|_2+g}{2}\right) & \text{if } \|x\|_2 > g, \|x\|_2 + g > 0 \\ (0, 0) & \text{if } \|x\|_2 > g, \|x\|_2 + g \leq 0 \end{cases} \tag{4.12}$$

To obtain the optimal parameter values different search methods have been proposed in (Schmidt, 2010). Here the Projected Quasi-Newton (PQN) optimization method is going to be employed. This method builds a second-order approximation of the objective function arround the current point, to find the direction minimizing the objective function. This method avoids the evaluation of the objective function in the neighborhood, assuming that the comptuation of projections is cheaper than the evaluation of the objective function. Readers are refered to the original publication for further details on the method.



4.3 Experimental evaluation

This section presents experimental evidency about the improvements that the overlapping group-regularized training of the HCRF produces in the accuracy of the trained models.

4.3.1 Experimental setup

Experiments are going to be conducted employing Weizmann Dataset (see appendix B.1 for more information on this dataset) The evalution over Weizmann dataset is done employing the 3072 dim descriptor containing the distance transform shown on appendix C.2.

The models to be tested in order to evaluate the proposal are.

1. HCRF: The standard HCRF model as shown on section 4.1, employing L2 regularization. Optimal model parameters are obtained with LBFGS optimization.

2. MFS-HCRF: The Hidden Conditional Random Field trained with L1 group regularization to perform feature and model selection, as shown in section 4.2.

The non-convexity of the loss functions employed to train these models forces the employment of a monte-carlo approach to evaluate every single configuration. The

obtained metrics are averaged over 30 trainings of each setup starting from different random initializations.

The different models are going to be trained employing $|\mathcal{H}| = 20$ hidden parts, $2\times$ the number of action classes in Weizmann dataset.

4.3.2 Experiment I: Choosing the right regularization parameter

The first experiment to be conducted is to select the optimal regularization parameter λ for each one of the models. The optimal regularization parameter is defined as the one minimizing the median negative log-likelihood obtained in the prediction of a set of test samples. Sequences from actors 2-9 are employed to train the model, while the sequences from actor 1 are employed as test set.

Boxplots on figures 4.3 and 4.4 respectively show the negative conditional likelihood (see appendix E.2) obtained by each one of the models in the prediction of Weizmann dataset. The negative log-likelihood values achieved by the MFS-HCRF model are smaller than the achieved by HCRF model. This indicates that the MFS-HCRF has a better predictive performance than the HCRF. The smaller negative log-likelihood value indicates that the MFS-HCRF produces more exact inferences than the HCRF when samples not available during training are presented. This fact confirms that incorporating model and feature selection to the HCRF improves predictive accuracy.

4.3.3 Experiment II: Action Recognition Results

Previous experiment has shown that the MFS-HCRF has better predictive accuracy than the HCRF for a single acotr. Now we conduct experiments to predictive the whole Weizmann dataset. LOAO-CV (see appendix E.1) is employed as the evaluation protocol. The regularization parameter λ for each one of the models is adjusted to the best value found in previous experiment.

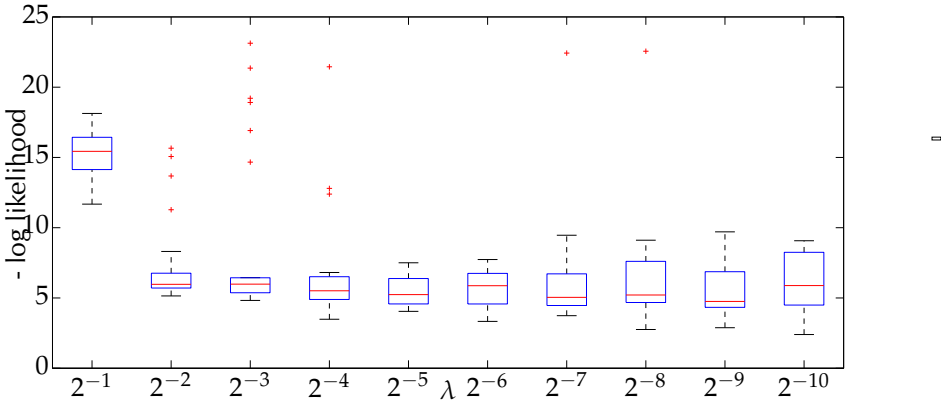


Figure 4.3: Negative log-likelihood values achieved with different values of λ training the HCRF model

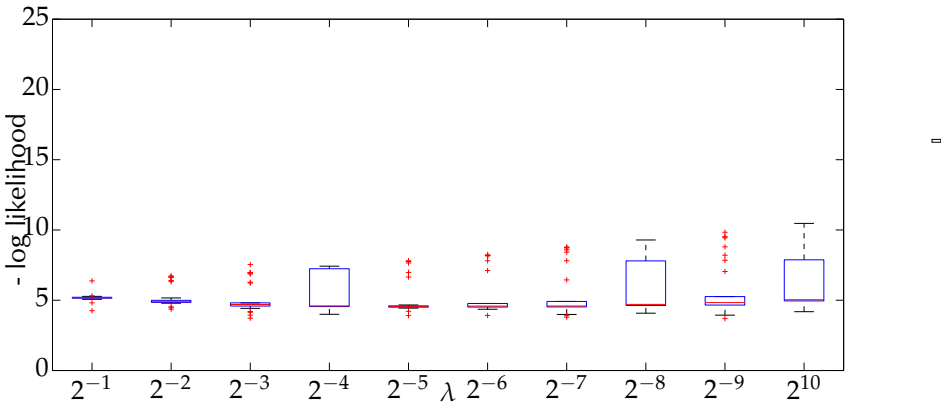
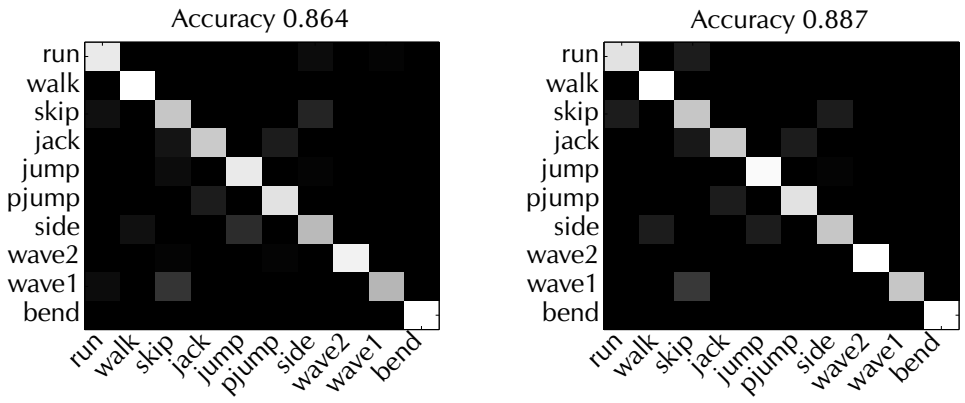


Figure 4.4: Negative log-likelihood values achieved with different values of λ training the MFS-HCRF model

Figures 4.5a and 4.5b shows the predictive performance achieved by both models. The MFS-HCRF model has a performance about a 2% higher than the HCRF in the prediciton of the whole dataset. Note that this results are far from the best reported for Weizmann dataset. It has been reported a perfect classification in works such (Gorelick *et al.* , 2007). The objective of the experiments here was to show that when training a HCRF with the proposed algorithm the obtained model has a better predictive performance than one trained with the standard algorithms.



(a) HCRF (b) MFS-HCRF

Figure 4.5: Confusion matrices obtained for the different models in the prediction of Weizmann dataset

4.4 Remarks

This chapter has presented an overlapping group-L1 regularization strategy to recover optimal HCRF sequence classifier parameters from a set of training samples. Model and feature selection is performed in the HCRF, reducing model overfitting. Experimental evaluation has shown that models trained with the proposed strategy have a higher predictive performance than those trained with the standard procedure.

5

Multiple View Learning for Human Action Recognition

Four eyes are better than two

Proberb

THE importance of multiple camera human action recognition systems has been already introduced in chapter 3. Robustness to occlusions, viewpoint invariance and wider scene coverage have been argued as some of the advantages of incorporating multiple cameras to human action recognition systems. The purpose of this dissertation is to design efficient algorithms to be deployed in Visual Sensor Networks, minimizing bandwidth usage and computational complexity to fulfill the imposed restrictions.

This chapter presents a first alternative to design efficient algorithms to perform human action recognition with multiple cameras. Dimensionality reduction was shown as an important step in action recognition in chapter 2. A well-known gdimensionality reduction framework is going to be extended to consider multiple views of the data, corresponding to motion descriptors computed from the multiple cameras observing the scene. Different algorithms are going to be defined as instantiation of the new proposed framework. FEI-FEO data fusion systems are going to be build to perform human action recognition. They have low computational complexity as

they rely on linear projections of the computed motion descriptors. The bandwidth usage is minimized as low dimensional descriptors will be sent over the network.

The chapter first presents the graph embedding framework in section 5.1. Then, the extension to multiple views is proposed in section 5.2. The computational issues related to obtaining the solutions of the framework are reviewed in section 5.3, presenting some approximation methods. The application of the proposed algorithm to predict humam actions from multiple cameras is presented on section 5.4. Chapter finishes remarking the main attributes of the proposed algorithms 5.5.

5.1 The Graph Embedding Framework

The Graph Embedding framework (Yan *et al.* , 2007) offers an unified view to understand and explain many dimensionality reduction algorithms, such PCA (Pearson, 1901), Isomap (Tenenbaum *et al.* , 2000), or LPP (Niyogi, 2004). High dimensional data is represented as the vertices of a graph, where the edges encode some statistical or geometrical property of the data. The graph is transformed to obtain low dimensional representation of the data preserving the encoded properties. Although the graph embedding framework abstracts both supervised and unsupervised dimensionality reduction algorithms, here only the unsupervised case is considered, without loss of generality.

Let $X = [x_1, x_2, \dots, x_N]$,$x_i \in \mathcal{R}^m$ be the matrix of N high dimensional zero mean data samples, and let $Y = [y_1, y_2, \dots, y_N]$,$y_i \in \mathcal{R}^{m'}$ be the low dimensional representations of the columns of X, $m' << m$. However, to simplify the exposition it is assumed that Y is one dimensional i.e., $y_i \in \mathcal{R}$. The objective of dimensionality reduction algorithms and thus, of the graph embedding framework, is to find a mapping function $F : \mathcal{R}^m \rightarrow \mathcal{R}^{m'}$ to transform the high dimensional data X in their low

dimensional representation Y:

$$Y = F(X) \tag{5.1}$$

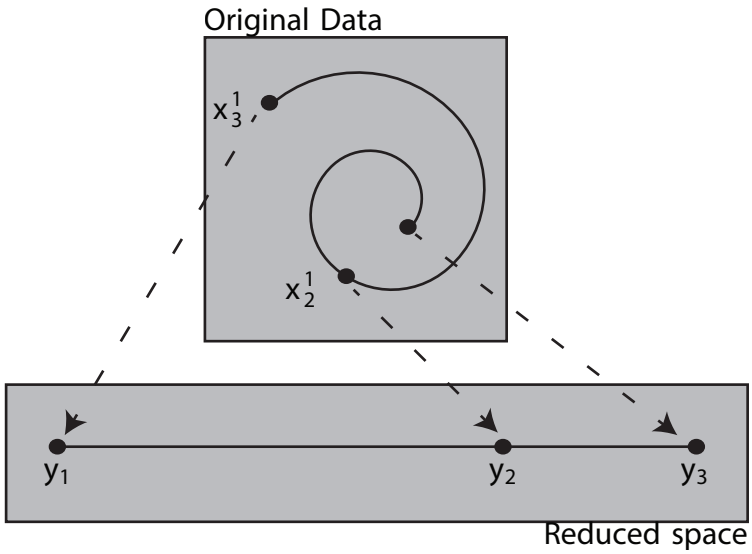


Figure 5.1: Example of the application of a dimensionality reduction algorithm

Figure 5.1 presents an example of dimensionality reduction from a 2D space to a 1D space. The variations in the spiral data in the original space only have a single degree of freedom. Thus, it is possible to map the spiral points to a line encoding the factor of variation.

The function F might be defined as an implicit mapping associating each point in the training set defined in $\mathcal{R}^m$ to a point in $\mathcal{R}^{m'}$; or be an explicit function to transform every point from $\mathcal{R}^m$ to $\mathcal{R}^{m'}$. In any case, the framework builds F from a graph defined in the high dimensional space. Let $G = \{X, W\}$ be an undirected weighted graph whose vertices are indexed by the columns of X and with edges weighted according to the values of the real simetric similarity matrix $W \in \mathcal{R}^{N \times N}$, where each component $W_{ij} = W_{ji}$ measures the similarity of the data at vertex i and the data at vertex j. Depending on the similarity measure employed different algorithms are derived from the framework, as it will be shown later.

Next paragraphs show the derivation of different solutions for the graph embedding framework: the implicit embedding, the linear projection and the kernel projection. The solutions are incrementally built from the implicit embedding.

5.1.1 Implicit Embedding

The first approach to obtain the optimal low dimensional representation Y of the data X is given by the optimal solution to the low dimensional embedding of the graph G:

$$Y = \arg \min_{Y^T BY = d} \sum_{i \neq j} \|y_i - y_j\|^2 W_{ij} = \arg \min_{Y^T BY = d} Y^T LY \tag{5.2}$$

where d is a constant and the matrix B is a restriction matrix to avoid the trivial solution of the objective function setting $Y = 0$. The matrix B is tipically de identity matrix I_N, but some algorithms impose harder restrictions. The Laplacian matrix L of the graph G is defined as:

$$L = D - W \tag{5.3}$$

where D is a diagonal matrix obtained by the sum of the values in the rows of W without the diagonal:

$$D_{ii} = \sum_{j \neq i} W_{ij} \forall i \tag{5.4}$$

The optimization problem in 5.2 finds low dimensional representations Y preserving the pairwise similarities between the vertices of the graph G defined by W. It has been shown (Cai *et al.* , 2007b) that it can be reformulated as an equivalent

maximization problem without the need of computing the graph Laplacian L.

$$Y = \arg \max_{Y^T B Y = d} Y^T W Y \tag{5.5}$$

The solution to the optimization problem in equation 5.5 is obtained solving the generalized eigenvalue problem:

$$Wy = \lambda By \tag{5.6}$$

Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m \geq 0 \in \mathcal{R}$ be the m different solution eigenvalues with corresponding eigenvectors $y_1, y_2, \ldots, y_m$ of the eigenproblem above. The m'-dimensional embedding of the graph G is given by the concatenation of the first m' eigenvectors: $Y = [y_1 y_2 \ldots y_{m'}]$.

This approach provides an implicit mapping from the high dimensional space $\mathcal{R}^m$ to the low dimensional space $\mathcal{R}^{m'}$. The main drawback of this formulation is that it does not define an explicit mapping function from the high dimensional space $\mathcal{R}^m$ to the low dimensional space $\mathcal{R}^{m'}$, needed to transform new high dimensional points not available during training.

5.1.2 Linearization

The simplest way to obtain a mapping function from the high dimensional space $\mathcal{R}^m$ to the low dimensional space $\mathcal{R}^{m'}$ is to define it as a linear mapping $Y = X^T \xi$ Where $\xi \in \mathcal{R}^m$ is a projection direction. Thus, the optimization problem in equation 5.2 is transformed to:

$$\begin{aligned} \xi &= \arg \min_{\xi^T X B X^T \xi = d} \sum_{i \neq j} \left\| \xi^T x_i - \xi^T x_j \right\|^2 W_{ij} = \arg \min_{\xi^T X B X^T \xi = d} \xi^T X L X^T \xi \\ &= \arg \max_{\xi^T X B X^T \xi = d} \xi^T X W X^T \xi \end{aligned} \tag{5.7}$$

The projection directions are obtained as the solutions to the generalized eigenvalue problem:

$$XWX^T\xi = \lambda XBX^T\xi \tag{5.8}$$

Let $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_m \geq 0 \in \mathcal{R}$ be the m different solution eigenvalues with corresponding eigenvectors $\xi_1, \xi_2, ..., \xi_m$ of the eigenproblem above. The m'-dimensional linear projection matrix is given by the concatenation of the first m' eigenvectors: $\Xi = [\xi_1\xi_2 ... \xi_{m'}]$.

5.1.3 Kernelization

The linearity assumption in the function F can be sometimes very hard. A non-linear mapping function might be obtained applying the kernel trick to the optimization problem in 5.7. Kernel methods (Smola & Schölkopf, 1998) provide a procedure to transform linear algorithms based on inner products of the input data to non-linear, mapping the input data $x_i \in \mathcal{X}$, where $\mathcal{X}$ is some inner product space, to a high dimensional feature space $\phi(x_i) \in \mathcal{F}$:

$$\phi : x_i = (x_{1i}, ..., x_{mi}) \rightarrow \phi(x_i) = (\phi_1(x_i), ..., \phi_N(x_i)) \quad (m << N)$$

The original algorithm is applied in the high dimensional feature space $\mathcal{F}$, but the mapping from the input to the feature space is not explicitly made. Instead, the inner products in the input space $\mathcal{X}$ computed by the original algorithm are replaced by inner products in the feature space $\mathcal{F}$. This inner products are computed employing kernel functions. A kernel is a function $K(x, z)$, such that for all $x, z \in X$

$$K(x, z) =< \phi(x) \cdot \phi(z) > \tag{5.9}$$

To formulate the graph embedding framework in terms of inner products in the input space $\mathcal{R}^m$, the direction of projection might be defined as the projection of the data X into some direction α:

$$\xi = X\alpha \tag{5.10}$$

Introducing it into the optimization problem in equation 5.7 an algorithm depending just on inner products of the inputs is obtained:

$$\begin{aligned}\alpha^* &= \arg \min_{\alpha^T X^T XBX^T X\alpha = d} \alpha^T X^T XLX^T X\alpha = \arg \min_{\alpha^T KBK\alpha = d} \alpha^T KLK\alpha \\ &= \arg \max_{\alpha^T KBK\alpha = d} \alpha^T KWK\alpha\end{aligned} \tag{5.11}$$

where $K_{ij} =< \phi(x_i), \phi(x_j) >$ is the gram matrix of the input data. In this work we employ a radial basis kernel:

$$K(x, z) = e^{-\frac{1}{2\sigma^2}\|x-z\|^2} \tag{5.12}$$

where σ is a parameter controling the bandwidth of the gaussian.

The solution to the optimization problem in equation 5.11 is given by the solutions of the generalized eigenvalue problem:

$$KWK\alpha = \lambda KBK\alpha \tag{5.13}$$

Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m \geq 0 \in \mathcal{R}$ be the m different solution eigenvalues with corresponding eigenvectors $\alpha_1, \alpha_2, \ldots, \alpha_m$ of the eigenproblem above. The m'-dimensional projection matrix from the feature space $\mathcal{F}$ spanned by the kernel K to the low dimensional space $\mathcal{R}^{m'}$ is given by the concatenation of the first m' eigenvectors: $A = [\alpha_1 \alpha_2 \ldots \alpha_{m'}]$.

5.1.4 Relationship to Principal Component Analysis

Principal Component Analysis (PCA) (Pearson, 1901) and Kernel Principal Component Analysis (Schölkopf *et al.* , 1997) (KPCA) are special cases of the linearization and kernelization of the graph embedding framework. The Principal Component directions are defined as the solution eigenvectors of the eigenvalue problem:

$$C\xi = \lambda\xi \tag{5.14}$$

where C is the empirical covariance matrix of the data. It is straightforward to show that this optimization problem is the same that the one in equation 5.8 setting $W = W_{PCA} = \frac{1}{N} I_N$, as $C = XW_{PCA}X^T$.

In the case of KPCA the empirical covariance matrix of the data is computed in a feature space instead of in the input space. The optimization problem solved by KPCA is equivalent to the one in equation 5.13 setting $W = W_{PCA}$, as the covariance in the feature space is computed in a similar way as $C = KW_{PCA}K^T$

5.1.5 Relationship to Isomap / Isometric Projections

The Isomap (Tenenbaum *et al.* , 2000), Isometric Projection (IsoP) and Kernel Isometric Projection (KIsoP) (Cai *et al.* , 2007a) algorithms are dimensionality reduction algorithms finding low dimensional representations of the data best preserving the geodesic distances between pairs of data along the manifold. Geodesic distances are approximated employing a neighborhood graph, capturing the local manifold structure. In this thesis the K-neighbor criteria is employed to build neighborhood graphs. The neighborhood graph $NN = (X, D)$ has vertices indexed by the N training samples and edges weighted by:

$$D_{ij} = \begin{cases} \|x_i - x_j\|_2 & \text{if } x_j \in \text{Neigh}_K(x_i) \\ \infty & \text{otherwise} \end{cases} \tag{5.15}$$

where $\text{Neigh}_K(x_i)$ denotes the set of K nearest neighbors of the point x_i. Once the neighborhood graph has been built, with local distances D, the approximate geodesic distances D_G are obtained computing the shortest path between every pair of points employing the Dijkstra algorithm.

The optimization problem solved by Isomap is defined as:

$$y = \arg\min_y \|\tau(D_G) - \tau(D_Y)|_L \tag{5.16}$$

The matrix D_Y contains the Euclidean distance between every pair of points in the low dimensional space. The function $\tau(D_G) = -HSH/2$, with $H = I - \frac{1}{N}ee^T$ and $S_{ij} = D^2_{G_{ij}}$ transforms the geodesic distances into similarities. Thus, the optimal embedding is the one best preserving the similarities obtained from the geodesic distances in the projected space. The Isomap formulation is equivalent to set up the matrix $W = W_{Isomap} = \tau(D_G)$ in the direct embedding formulation of equation 5.5. Similarly, the formulation of IsoP and KIsoP correspond respectively to the linearization an kernelizations of the graph embedding framework in equations 5.7 and 5.11 setting W to $W = W_{Isomap}$

5.2 Multiview Graph Embedding

Previous section has presented the graph embedding dimensionality reduction framework. A contribution of this thesis is the extension of the graph embedding framework to the case with multiple views of the data. Multiple views of the data refers to data defined in multiple feature spaces with the same underlying information about an event to predict on them. The idea behind the extension is that the data on each view might be parameterized on a low dimensional manifold, and that the manifolds on each view are homeomorphic between them. Thus, it is possible to find a transformation producing similar low dimensional representations for each point at each view.

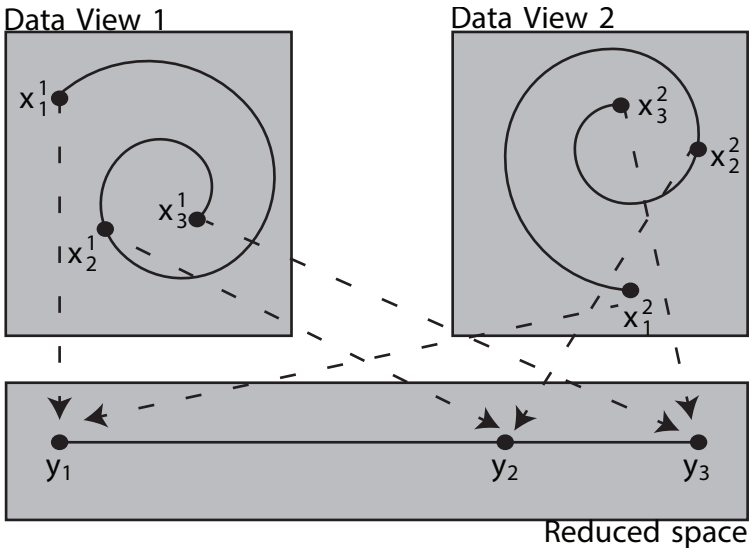


Figure 5.2: Example of multiple view dimensionality reduction

Figure 5.2 illustrates the application of the proposed framework. The spiral example shown on previous section is extended to the multiple view case, where the spiral data is rotated in the different views. However, the 1D representations obtained are similar.

Assume there is K different views of the high dimensional input dataset $X^k = [x_0^k, \cdots, x_N^k], x_i^k \in \mathcal{R}^{m^k}$. The objective is to find an embedding function $F : \mathcal{R}^{m^1} \times \ldots \times \mathcal{R}^{m^K} \rightarrow \mathcal{R}^{m'}$ to transform the data X to a low dimensional space.

$$Y = F\left(X^1, \ldots, X^K\right) \tag{5.17}$$

This function is going to be decomposed as the sum of invidual transformations of the data obtained from each view:

$$Y = \sum_{i=1}^{K} F^i(X^i) \tag{5.18}$$

Thus, a different embedding Y^i is obtained for each one of the views of the high

dimensional data X^i:

$$Y^i = F^i\left(X^i\right) \tag{5.19}$$

The functions $F^1, \ldots, F^K$ are going to be jointly derived following the graph embedding framework formalism, obtaining implicit embedding, linear projection and kernel projection formulations.

The derivation starts with the definition of the graph G. Let $G = \left(\bigcup_{1\leq k\leq K} X^k, \bigcup_{1\leq i,j\leq K} W^{ij}\right)$ be an undirected weighted graph with vertices indexed by the union of the K views of the data, and edges weighted by the set of weight matrices W^{ij}, denoting intraview similarities when $i = j$ and interview similarities when $i \neq j$. Again, different forms of measuring similarity between pairs of data will lead to different algorithms, as it will be shown later.

The implicit embedding, linear projection and kernel projection solutions of the graph embedding framework are derived for the multiple view case in next sections, employing the new definition of the graph G. The derivations are similar to the described on previous section.

5.2.1 Implicit embedding

The implicit embedding formulation for the different views of the data is defined as the solution to the optimization problem:

$$\begin{aligned} Y &= \arg\min \sum_{\forall k,l} \sum_{i\neq j} \left\| y_i^k - y_j^l \right\| w_{ij}^{kl} = \arg \min_{Y^T BY = d} Y^T LY \\ &= \arg \max_{Y^T BY = d} Y^T WY \end{aligned} \tag{5.20}$$

where $Y = \left[Y^1 \ldots Y^K\right]$ denotes the concatenation of the embedded components for the different views of the data. The matrix W is obtained by the concatenation of

the interview and intraview submatrices:

$$W = \begin{bmatrix} W^{11} & \dots & W^{1K} \\ \vdots & \ddots & \vdots \\ W^{K1} & \dots & W^{KK} \end{bmatrix} \tag{5.21}$$

The low dimensional representations for the different views are obtained solving the eigenproblem in equation 5.6, as in the single view case.

5.2.2 Linearization

The linearization of the multiple view graph embedding is made defining the explicit mapping $Y = X^T\xi$, where $X = \begin{bmatrix} X^1 & & \\ & \ddots & \\ & & X^K \end{bmatrix}$ is defined as a block diagonal matrix containings the different views of the data, and $\xi = \begin{bmatrix} \xi^1 \\ \vdots \\ \xi^K \end{bmatrix}$ is the concatenation of the projection directions ξ^k for the K different views of the data:

$$\begin{aligned} \xi &= \arg\min \sum_{\forall k,l} \sum_{i \neq j} \left\| \xi^k x_i^k - \xi^l x_j^l \right\| w_{ij}^{kl} = \arg \min_{\xi^T X B X^T \xi = d} \xi^T X L X^T \xi \\ &= \arg \max_{\xi^T X B X^T \xi = d} \xi^T X W X^T \xi \end{aligned} \tag{5.22}$$

The projections directions are obtained as the solution to the generalized eigenvalue problem in equation 5.8, plugging the new definitions of X and W.

5.2.3 Kernelization

The kernelization of the multiple view direct graph embedding is made defining each vector $\xi^k = X^k \alpha^k$ as the projection of the data at view k into some direction α^k. Plugging this definition into equation 5.22 leads to the optimization problem:

$$\begin{aligned} \alpha^* &= \arg \min_{\alpha^T X^T X B X^T X \alpha = d} \alpha^T X^T X L X^T X \alpha = arg \min_{\alpha^T K B K \alpha = d} \alpha^T K L K \alpha \\ &= \arg \max_{\alpha^T K B K \alpha = d} \alpha^T K W K \alpha \end{aligned} \tag{5.23}$$

where $K = \begin{bmatrix} K^{11} & & \\ & \ddots & \\ & & K^{KK} \end{bmatrix}$ is the block-diagonal matrix with the kernel matrices obtained at each one of the views of the data.

Again, the projection directions α are solved plugging the new definitions of K and W into the generalized eigenvalue problem described at equation 5.13.

5.2.4 Relationship to Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) (Hardoon *et al.* , 2004) is a multiple view dimensionality reduction method finding set of projections maximizing the correlation among the transformed variables. The objective of CCA is to find a pair of linear projections maximizing the correlation in the projected space between a pair of multivariate random variables . Given the zero mean random variables in the input space X^1 and X^2 with dimensions m^1 and m^2, CCA finds a pair of linear transformations ξ_1, ξ_2, such that one component within each set of transformed variables is correlated with a single component in the other set. The correlation between the corresponding components is called canonical correlation, and there can be at most $d = min(d_1, d_2)$ canonical correlations. The first canonical correlation is defined as:

$$\rho = \max_{\xi_1,\xi_2} \frac{\left\langle \xi_1^T x_1 \cdot \xi_2^T x_2 \right\rangle}{\sqrt{\left\langle \|\xi_1^T x_1\|^2 \right\rangle \left\langle \|\xi_2^T x_2\|^2 \right\rangle}} \tag{5.24}$$

$$= \max_{\xi_1,\xi_2} \frac{\xi_1^T \left\langle x_1 x_2^T \right\rangle \xi_2}{\sqrt{\xi_1^T \left\langle x_1 x_1^T \right\rangle \xi_1 \xi_2^T \left\langle x_2 x_2^T \right\rangle \xi_2}} \tag{5.25}$$

where $\left\langle x_1 x_1^T \right\rangle$, $\left\langle x_2 x_2^T \right\rangle$ and $\left\langle x_1 x_2^T \right\rangle$ are estimated as $\tilde{\Sigma}_{11}$, $\tilde{\Sigma}_{22}$ and $\tilde{\Sigma}_{12}$ respectively, i.e, the different minors of the empirical covariance matrix $\tilde{\Sigma} = \begin{pmatrix} \tilde{\Sigma}_{11} & \tilde{\Sigma}_{12} \\ \tilde{\Sigma}_{21} & \tilde{\Sigma}_{22} \end{pmatrix}$ of a set of training data $x = (x_1, x_2)$. The remaining canonical correlation directions are orthogonal to ξ_1 and ξ_2 respectively. They are computed as the solutions of the generalized eigenvalue problem:

$$\begin{pmatrix} \tilde{\Sigma}_{11} & \tilde{\Sigma}_{12} \\ \tilde{\Sigma}_{21} & \tilde{\Sigma}_{22} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} = (1+\rho) \begin{pmatrix} \tilde{\Sigma}_{12} & 0 \\ 0 & \tilde{\Sigma}_{21} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix}$$

The standard CCA model is defined for only two random variables x_1 and x_2. Bach and Jordan (Bach & Jordan, 2003) generalize it to K random variables. The generalized eigenvalue problem to solve is defined as:

$$\begin{pmatrix} \tilde{\Sigma}_{11} & \cdots & \tilde{\Sigma}_{1K} \\ \vdots & & \vdots \\ \tilde{\Sigma}_{K1} & \cdots & \tilde{\Sigma}_{KK} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \vdots \\ xi_K \end{pmatrix} = \lambda \begin{pmatrix} \tilde{\Sigma}_{11} & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & \tilde{\Sigma}_{KK} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \vdots \\ \xi_K \end{pmatrix}$$

where $\begin{pmatrix} \tilde{\Sigma}_{11} & \cdots & \tilde{\Sigma}_{1K} \\ \vdots & & \vdots \\ \tilde{\Sigma}_{K1} & \cdots & \tilde{\Sigma}_{KK} \end{pmatrix}$ denotes the empirical covariance matrix of a set of training data $x = (x_1, \ldots, x_K)$

This optimization problem is equivalent to the lineariztion of the multiple view graph embedding framework with the matrix W set to $W = W_{CCA} = \begin{bmatrix} \frac{1}{N}I_N & \dots & \frac{1}{N}I_N \\ \vdots & \ddots & \vdots \\ \frac{1}{N}I_N & \dots & \frac{1}{N}I_N \end{bmatrix}$ and the matrix $B = \frac{1}{N}I_{KN}$

Following the same reasoning it can be shown that the formulation for Kernel Canonical Correlation Analysis (KCCA) (Bach & Jordan, 2003) is equivalent to the kernelization of the multiple view graph embedding framework with the same definitions of the W and B matrices.

5.2.5 Relationship with the Joint Manifolds Framework

An extension of Isomap for multiple view dimensionality reduction has been proposed by (Davenport *et al.* , 2010) with the general idea of obtaining the joint parameterization of the data into the same low dimensional manifold.

Let D^k denote the matrix of Euclidean distances of the data at view k. The work defines the matrix $D = \sum_{k=1}^{K} D^k$ and runs the Isomap algorithm employing D as input to obtain the embeddings of the data $Y^1 = Y^2 = \dots = Y^k$. The solution proposed in their work is equivalent to set the matrix W in the direct embedding framework of equation 5.20 to:

$$W_{JMIsoMap} = \begin{bmatrix} \tau(D_G) & & \\ & \ddots & \\ & & \tau(D_G) \end{bmatrix} \tag{5.26}$$

Note that the solution eigenvectors of the eigenproblem that arises is the concatenation of the eigenvectors of $\tau(D_G)$, resulting in a simpler eigenvalue problem. The linearization and kernelization of their proposal is obtained plugging $W_{JMIsoMap}$ into equations 5.22 and 5.23. These method will be respectively denoted as JMIsoP and JMKIsoP in subsequent sections.

5.2.6 Multiview Isomap, Multiview Isometric Projections and Multiview Kernel Isometric Projections

Previous section has shown an extension of Isomap to the case of multiple views of the data. Here an alternative is explored, based on the definition of graphs connecting the representations of the data between the different views of the data. It is possible to define these weights in multiple ways. Here the distances between the representation of each sample at the different views is defined to be zero, as ideally the representation in the reduced space of a point in the different views should be zero:

$$\left\| x_i^k - x_i^l \right\|_2 \doteq 0 \quad \forall i, k, l \tag{5.27}$$

This definition is employed to build the k-nearest neighbor graph of each view. The distance between a pair of points is defined to be the minimum distance across the different views:

$$D_{ij} = \left\| x_i^l - x_j^l \right\|_2 \doteq \min \left(\left\| x_i^1 - x_j^1 \right\|_2, \dots, \left\| x_i^K - x_j^K \right\|_2, \right) \quad \forall l \tag{5.28}$$

This implies that the intraview and interview distances are all the same. Thus, the matrix W is defined as:

$$W = \begin{pmatrix} \overbrace{\begin{matrix} \tau(D_G) & \dots & \tau(D_G) \\ \vdots & \ddots & \vdots \\ \tau(D_G) & \dots & \tau(D_G) \end{matrix}}^{K} \Bigg\} K \end{pmatrix} \tag{5.29}$$

Pluggin the definition of W into optimization problems for the implicit embedding, linearization and kernelization will lead to algorithms denoted in subsequent sections as MVIsomap, MVIsoP and MVKIsoP. The special structure of the matrix

W provides some computational advantages that are going to be exploited later to efficiently solve the problem.

5.3 Computational issues

Computing solutions to the graph embedding framework presented on section 5.1 is an ill possed problem in terms of the number of samples. The solution to the direct embedding formulation on equation 5.6 requires to compute the eigendecomposition of a - possibly - dense $N \times N$ matrix. The linearization formulation in equation 5.8 requires the multiplication of very large dense matrices and the eigendecomposition of a matrix of size $m \times m$. The Kernelization in equation 5.13 is even worse, as it needs to multiply $N \times N$ matrices and then compute the eigendecomposition of a matrix of size $N \times N$. Beyond temporal complexities, the problem for large N is that the matrices does not fit in memory, so it is not possible to apply in a direct way the framework to large scale learning problems. Fortunately, there exist different proposals to compute approximate solutions to the graph embedding framework for large N.

The multiview graph embedding framework makes the storage problem even worse, as the matrices involved in the calculus are of size $KN \times KN$. However, for some types of matrices the computations might be reduced to eigenproblems of size $N \times N$.

This section presents some computational tricks that has bee proposed to make tractable the computation of solutions of the graph embedding framework and, by extension, to the multiple view graph embedding framework. It is also covered the case when the W matrix in the multiple view framework has an special structure, as it is the case in JIsoMap/JIsoP/JKIsoP.

5.3.1 The Spectral Regression Framework

The spectral regression framework (Cai *et al.* , 2007b) has been proposed to transform the eigenproblems appearing in the linearization and kernelization formulations of the graph embedding framework relating their solution to the solution of the direct embedding problem. This way the multiplication of large - possibly dense - matrices is avoided.

A theorem appearing in (Cai *et al.* , 2007b) stands that if y is the eigenvector of eigenproblem in Eqn. (11) with eigenvalue λ then, if $X^T a = y$, a is the eigenvector of eigenproblem in Eqn. (12) with the same eigenvalue λ. If $K\alpha = y$, then α is the eigenvector of eigen-problem in Eqn. (13) with the same eigenvalue λ.

This theorem implies that the solutions of the linear and kernelized formulations of the graph embedding framework and, by extension, of the multiple view graph embedding framework, might be obtained with multivariate regression from the solution to the direct embedding formulation,

The solution in this way has some additional advantages, such the possibility of incorporating regularization to the graph embedding framework, leading to more robust solutions. The usage of sparse regularizers also leads to compact projections functions, less prone to overfitting and with a reduced computational cost. The study of this effects is outside the scope of this dissertation.

Note that this approximation is only possible when the matrix W does not have a trivial eigenvalue decomposition as is the case of PCA and CCA.

5.3.2 The Nystrom Approximation

The computation and storage of the full matrix $W \in \mathcal{R}^{N \times N}$ is not tractable. However, it is possible to compute and approximation of the eigenvectors and eigenvalues of W from an approximation matrix $\tilde{W}$ build from a set of $I << N$ columns. Let C be the $N \times I$ matrix with the sampled columns, and A be the $I \times I$ matrix with the

intersection of the sampled l columns with the corresponding l rows of G. Without loss of generality the matrix W might be rearranged such:

$$W = \begin{bmatrix} A & W_{21}^T \\ W_{21} & W_{22} \end{bmatrix} \tag{5.30}$$

$$C = \begin{bmatrix} A \\ W_{21} \end{bmatrix} \tag{5.31}$$

A popular method to build the approximation $\tilde{W}$ is the Nyström method. It has been employed to obtain approximate solutions of large scale kernel problems (Williams & Seeger, 2001), and proposed as a method to get approximate soltuions of the graph embedding framework (Talwalkar *et al.* , 2008). The Nyström approximation method defines the approximation $\tilde{W}$ as:

$$W \approx \tilde{W} = CA^{+}C^{T} \tag{5.32}$$

where A^{+} is the Moore-Penrose pseudoinverse of A. As the number of sampled columns l increases, $\tilde{W}$ converges to W . The approximate eigenvalues $\tilde{\Sigma}$ and eigenvectors $\tilde{U}$ of W are given by:

$$\tilde{\Sigma} = \frac{N}{L}\Sigma_W \tag{5.33}$$

$$\tilde{U} = \sqrt{\frac{L}{N}} C U_W \Sigma_W^{+} \tag{5.34}$$

where $W = U_W \Sigma_W U_W^T$.

This way the computational complexity of obtaining the top k eigenvalues and eigenvectors of W is reduced from $\mathcal{O}\left(N^3\right)$ to $\mathcal{O}\left(L^3 + kLN\right)$

The approximation implies that the graph weights should only be given from the

set of L sampled points to the N total points. In the case of the Isomap related algorithms where the Dijkstra algorithm should be employed to compute the geodesic pairwise distances the complexity is reduced from $\mathcal{O}\left(N^2 \log N\right)$ to $\mathcal{O}\left(LN \log N\right)$. However, the nearest neighbor graph should be first constructed with a cost of $\mathcal{O}\left(N^2\right)$.

5.3.3 Solving the Multiview Isomap problem

The matrix $W_{MVIsoMap}$ presented in section 5.2.6 has an special structure allowing the simple computation of their eigendecomposition. Here a theorem is derived to show how to do it.

Theorem 1. *Let A be a symetric semi-positive definite matrix with eigenvalues $\lambda_1^A \geq \lambda_2^A \geq \ldots \lambda_N^A \geq 0 \in \mathcal{R}$ and associated eigenvectors $\xi_1^A, \xi_2^A, \ldots, \xi_N^A$. Let B a block matrix defined by the concatenation of matrix A vertically and horizontally K times:*

$$B = \begin{pmatrix} A & \cdots & A \\ \vdots & \ddots & \vdots \\ A & \cdots & A \end{pmatrix} \tag{5.35}$$

B has at most N non-zero eigenvalues given by $\lambda_i^B = k\lambda_i^A$, $1 \leq i \leq N$. The eigenvectors of B are given by the concatenation of the eigenvectors of A K times:

$$\xi_i^B = \begin{bmatrix} \xi_i^A \\ \vdots \\ \xi_i^A \end{bmatrix}$$

Proof. The proof starts reasoning about the range of the matrix B. It is straightforward to show that $N \times (K-1)$ rows/columns of B are a linear combination of the remaining K rows/columns of B: they are the same. Thus, B has at most N non-zero eigenvalues.

The eigenvalues λ^A and eigenvectors ξ^A of the matrix A are defined as the solutions to the problem:

$$A\xi^A = \lambda\xi^A \tag{5.36}$$

The eigenvalues λ^B and eigenvectors ξ^B of the matrix B are defined as the solutions to the problem:

$$B\xi^B = \begin{pmatrix} A & \cdots & A \\ \vdots & \ddots & \vdots \\ A & \cdots & A \end{pmatrix} \begin{pmatrix} \xi^A \\ \vdots \\ \xi^A \end{pmatrix} = \lambda^B\xi^B \tag{5.37}$$

From here it might be stated that:

$$\sum_{k=1}^{K} A\xi^A = kA\xi^A = \lambda^B\xi^B = K\lambda^A\xi^A \tag{5.38}$$

That completes the proof. □

Employing this theorem, the solution to MVIsoMap is reduced from the eigendecomposition of a $KN \times KN$ matrix to the eigendecomposition of a $N \times N$ matrix.

5.4 Application: Multiple Camera Human Action Recognition

The performance of the multiple view dimensionality reduction algorithms instanciated from the multiple view graph embedding framework presented in this chapter is going to be evaluated in a multiple camera human action recognition task, employing the 5 camera views in the IXMAS dataset (see appendix B.2 for details). The application of the multiple view graph embedding framework for the recognition of

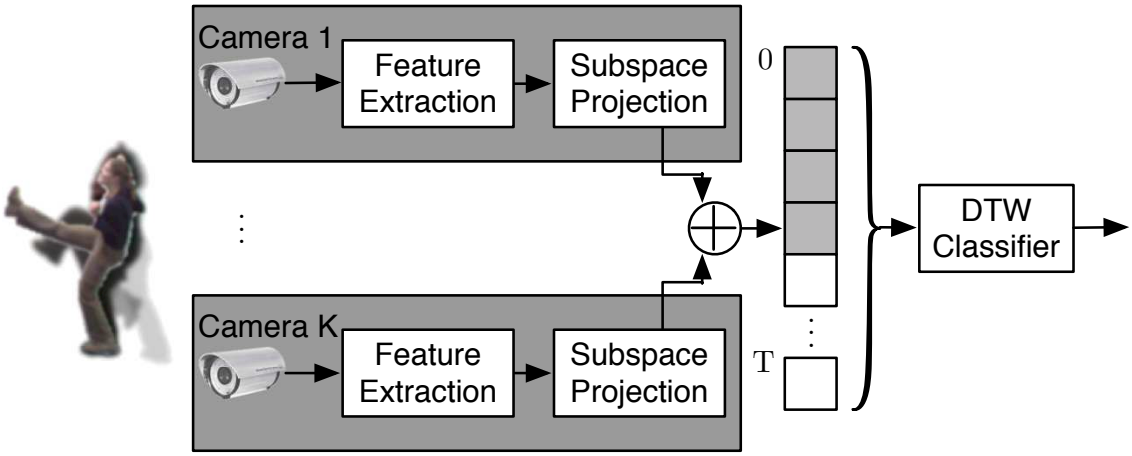


Figure 5.3: Strucuture of the system for human action recognition from multiple cameras build to evaluate the multiple view graph embedding framework

human actions from multiple cameras might be understood as a FEI-FEO data fusion method as the presented on chapter 3.

5.4.1 Experimental Setup

The system build to test the framework is shown on figure 5.3. For each one of the frames grabbed by the cameras in the system the motion descriptor proposed by (Tran & Sorokin, 2008) and described in appendix C.1 is extracted. The motion descriptor has a dimensionality $m = 286$. Linear and Kernel projections are learned to reduce the dimensionality of the motion descriptors. Different instantiations of the graph embedding framework are going to be employed: CCA, KCCA, JMISOP, JMKISOP, MVISOP and MVKISOP. The direct embedding formulations are not tested as they do not provide a way to transform unknown samples to the projected space. The learned subspaces are going to be learned with dimensionalities $m' = 10, 15, 20, 25$ to study the effect of their variation in the final result. Once the frames of each sequence have been projected they are introduced into a K-Nearest Neighbor classifier based on the Dynamic Time Warping (DTW) Distance. The classifier is going to be tested with $K = 5$ and $K = 10$ neighbors. This classifierhas been selected for the simplicity and speed of its usage. More complex classifiers are expected to have a

better performance. See appendix D for additional details on this classifier.

Due to the large number of samples ($N \approx 20000$), the Nyström approximation presented in section 5.3.2 is applied to obtain approximate solutions in the case of JMIsoP, JMKIsoP, MVIsoP and MVKIsoP with a subsample of $l = 3000$ points. The radial basis kernel is fixed with a parameter $\sigma = 0.5$.

The kernel ridge regression for MVKIsoP and JMKIsoP is not tractable, and it is going to be approximated employing the Nyström methods as proposed in (Talwalkar, 2010) with $L = 1000$ samples. Regularization parameters for the ridge regression are set to $\lambda = 0.1$. Experimental evidence has shown that the final results are not very sensitive to small perturbations of this value.

The accuracy of the system is going to be evaluated employing Leave One Actor Out Cross-Validation. See Appendix E.1 for additional information. The performance of every fusion model involving sampling - all except CCA - is measured 30 times following a Monte-Carlo approach.

Finally, to measure the improvement produced by the fusion method in the predictive performance, a baseline model is going to be employed. Actions are going to be predicted with the same experimental configuration but employing data from a single camera. PCA is going to be employed to compute the projections.

5.4.2 Results

In order to visualize the effect of the feature fusion algorithms, the three most significant features obtained by CCA and PCA baseline are shown on figures 5.4-5.5. The results obtained by other methods are quite similar to the obtained by CCA and are thus ommited for space reasons. The projections have been obtained with the data of actors 2-10. It can be observed that the fused features have a stroger class structure than the features obtained by the PCA baselines, although they do not seem to be separated in any case. This is something normal in the action recognition domain, as the different action sequences usually share common frames, being their temporal

evolution the real discriminative factor to predict action classes. Another surprising result is the similar class structure that the fused features have independently of the fusion method employ.

d	CCA		KCCA		JMIsoP		JMKIsoP		MVIsoP		MVKIsoP	
	k=5	k=10	k=5	k=10	k=5	k=10	k=5	k=10	k=5	k=10	k=5	k=10
10	.8889	.8949	.8936	.879	.7283	.0.7269	.7485	.7522	.8768	.8705	.8868	.8682
15	.8859	.8679	.8943	.8902	.8572	.8515	.8661	.8640	.8958	.8893	.8858	.88418
20	.8919	.8799	.8934	.8986	.8897	.8791	.8921	.8815	.9049	.9068	.8968	.89299
25	.8799	.8739	.9016	.8959	.8918	.8793	.9012	.8922	.9115	.9063	.9098	.91261

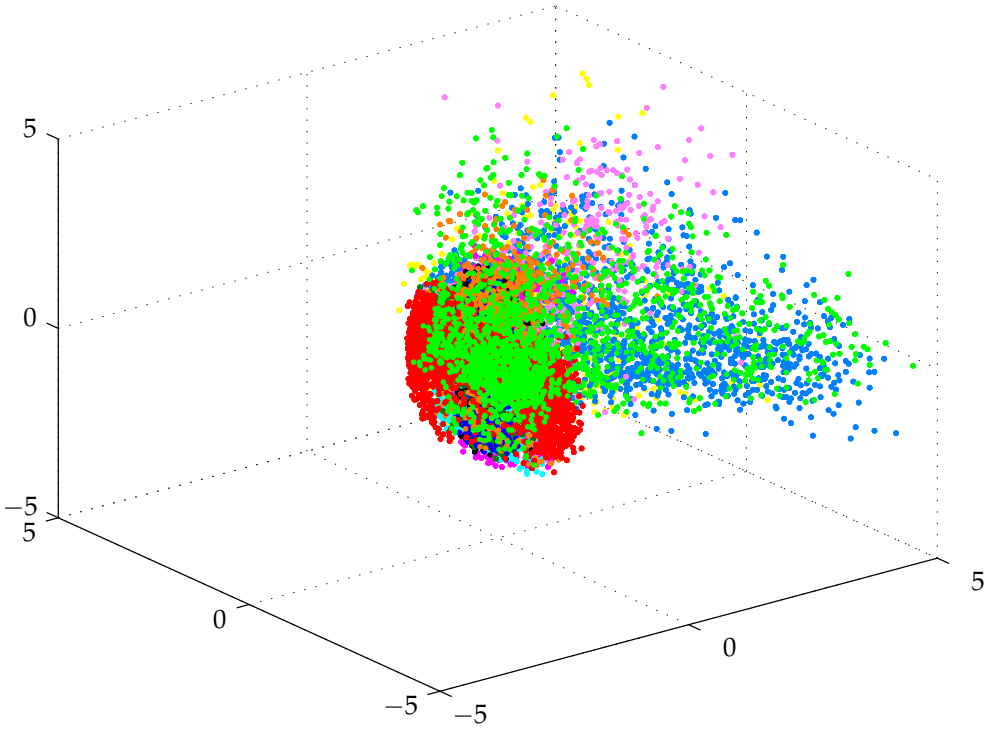
Table 5.1: Results obtained by the fusion methods

d	CAM 1		CAM 2		CAM 3		CAM 4		CAM 5	
	k=5	k=10	k=5	k=10	k=5	k=10	k=5	k=10	k=5	k=10
10	.6066	.6126	.6577	.6186	.5736	.5345	.6066	.5916	.7297	.7297
15	.6246	.6156	.6096	.5766	.5976	.5586	.6637	.6156	.7417	.7117
20	.6096	.5616	.6126	.5676	.6036	..5736	.6577	.6126	.7147	.6456
25	.5829	.5522	.6034	.5522	.5836	.5422	.6486	.5946	.6817	0.6006

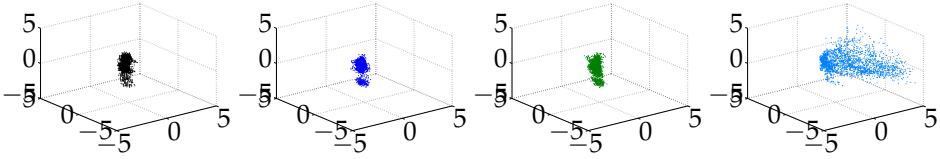
Table 5.2: Results obtained by the PCA baseline for each one of the cameras

Table 5.2 shows the accuracy achieved by the baseline method for each one of the cameras, and table 5.1 shows the accuracy achieved after employing the fused features. An increase of about a 20% in the accuracy is observed. The proposed multiple view extension of Isomap, MVIsoP achieves the best performance on the task, although the overall performance of all the methods is very similar. This might be caused because the information shared by the different descriptors is easy to extract and it is not necessary to employ very complex methods. Confusion matrices for the best classifiers found for each fusion method are shown on figure 5.6. It can be observed that the different classifiers fail discriminating between classes 'wave' and 'scratch head'. We think this failure is motivated by the feature extraction method employed, as we have observed this phenomena in other experiments employing the same feature vector.

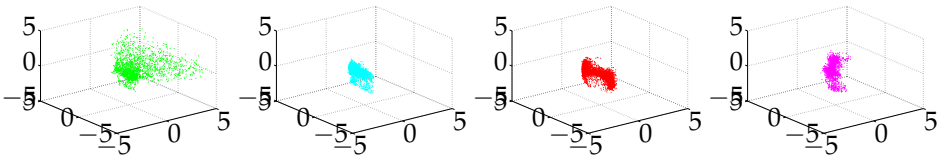
Finally, table 6.4 compares the results of the presented proposal to others. The presented approach compares good to other multicamera human action recognition methods applied to the Ixmas dataset, achieving an accuracy a bit smaller than methods employing 3D features. Note that these works employ HMM classifiers for the prediction, probably producing results better than ours.



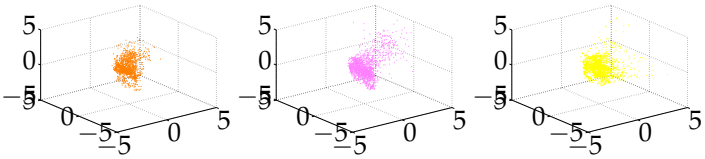
(a) 11 Action classes



(b) Check watch (c) Cross arms (d) Scratch head (e) Sit down

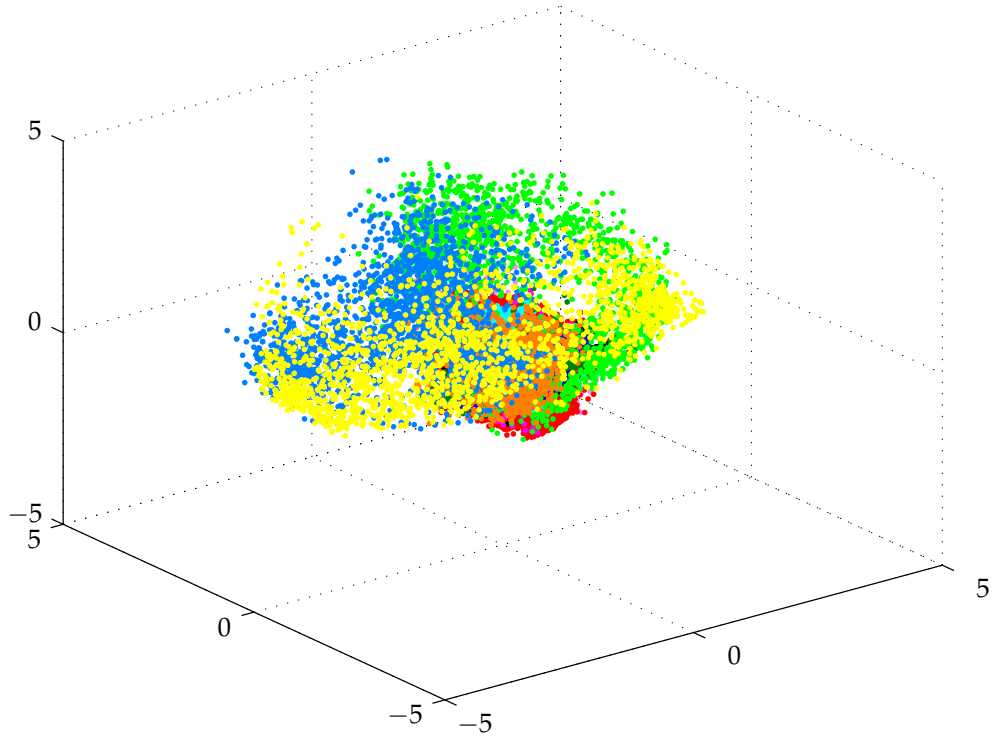


(f) Get up (g) Turn around (h) Walk (i) Wave

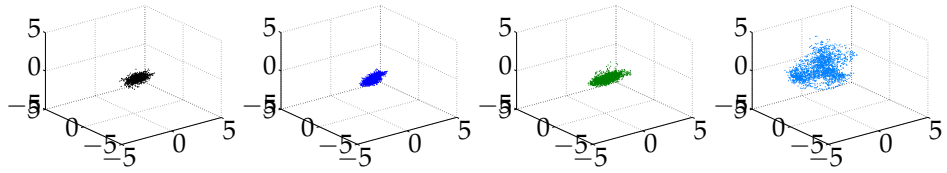


(j) Punch (k) Kick (l) Pick Up

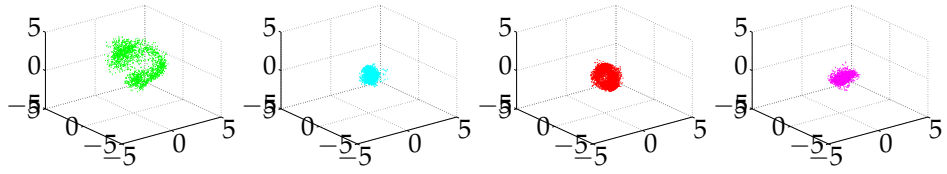
Figure 5.4: Projection of the 3 most significant features obtained using PCA of camera 1 data



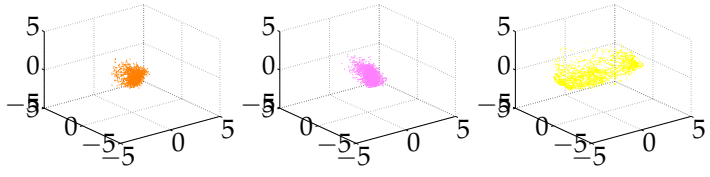
(a) 11 Action classes



(b) Check watch (c) Cross arms (d) Scratch head (e) Sit down



(f) Get up (g) Turn around (h) Walk (i) Wave



(j) Punch (k) Kick (l) Pick Up

Figure 5.5: Projection of the 3 most significant features obtained using CCA of camera 1 data

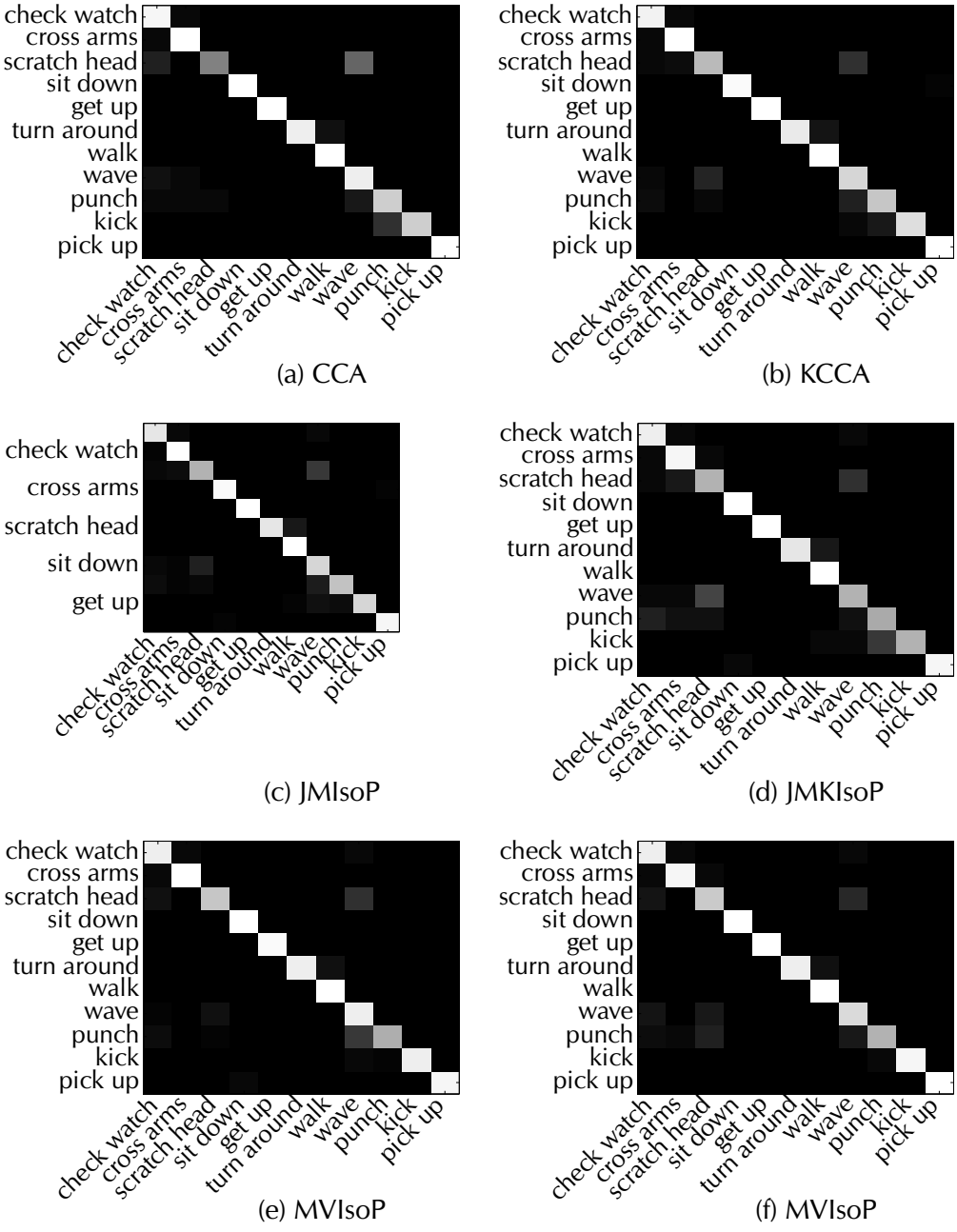


Figure 5.6: Confusion matrices for the best classifiers found for each one of the data fusion methods

Method	Accuracy	Type
Srivastava et al. (Srivastava *et al.* , 2009)	81.4	Decision-in Decision-out
Our's Best	91.26	2D Feature-in Feature-out
Weinland et al. (Weinland *et al.* , 2006)	93.33	2D Feature-in 3D Feature-out
Peng et al. (Peng *et al.* , 2009)	94.59	2D Feature-in 3D Feature-out

Table 5.3: Comparison of the accuracy of our method to others

5.5 Remarks

This section has proposed an extension of the graph embedding framework to deal with data defined in multiple feature spaces. It has been shown that different multiple view dimensionality reduction algorithms already existent are special cases of the proposed framework. A new multiple view dimensionality reduction algorithm has been developed. The framework has been applied to the recognition of human actions from multiple cameras. The validity of the proposed model has been shown predicting IXMAS dataset with an accuracy similar to the reported by 3D visual hull models.

6

Decision fusion for Human Action Recognition

The whole is greater than the sum of its parts

Metaphysica. Aristotle

PREVIOUS chapter has presented a multiple view dimensionality reduction framework that was employed as a FEI-FEO data fusion method for multiple camera human action recognition. This chapter presents an alternative approach for the recognition of human actions from multiple cameras that is located at the DEI-DEO data fusion level of Dasarathy's hierarchy (see chapter 3). The proposal in previous assumed that all the cameras provide the same information to predict action class label. However, that assumption is probably false. Probably a camera is better to predict some actions while others are better predicted by other cameras. The DEI-DEO fusion approach in this chapter wants to handle that uncertainty in order to explore an alternative approach.

6.1 System overview

Figure 6.1 illustrates the proposed multicamera action recognition architecture. K different cameras observe a scene from different viewpoints. It is assumed that there

is only a single individual in the scene. This way, we can ignore data for tracking association problems. Without loss of generality, it is also assumed that all K cameras always have a perception of the individual in the scene, although the number of cameras observing the individual may be different at every instant t. This should simplify ongoing formulations. The goal of the system is to select the action α performed by the individual from a set of N predefined actions $A = (a_1, \ldots, a_N)$ known a priori given a set of image sequences $\{I(x, y, t)^k\}$, $1 \leq t \leq T$, $1 \leq k \leq K$, of length T simultaneously acquired by the K cameras observing the scene.

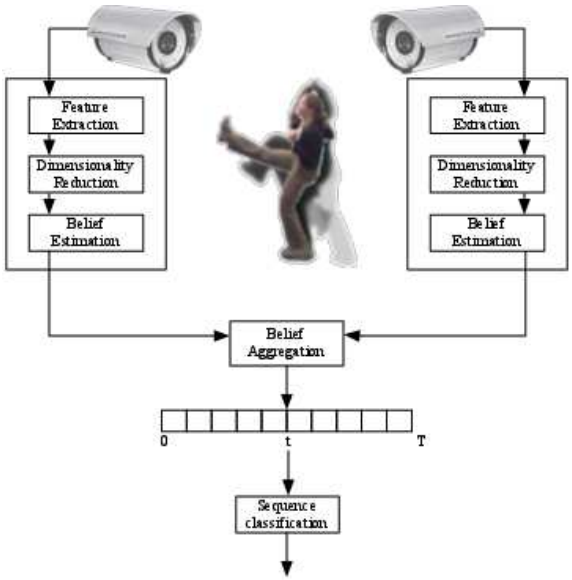


Figure 6.1: Overview of the proposed system

The first step in order to make this decision is to compute an action descriptor $f_t^k \in \mathcal{X}$ from the data grabbed from each view k. $\mathcal{X}$ is the inner product space where the descriptor is defined and typically $\mathcal{X} \equiv \mathcal{R}^D$, although other choices are also possible, for example when using histogram descriptors (Chaudhry *et al.* , 2009). f_t^k must capture enough variability in the data to be able to differentiate the actions in A. Another desirable property is that $\mathcal{X}$ should be compact in order to overcome the problems caused by the *curse of dimensionality*. Dimensionality reduction methods as the introduced in chapter 4 might be employed to project the original descriptors into a more compact subspace, if necessary.

Once an action descriptor f_t^k has been obtained, a probabilistic classifier is used to create a posterior probability distribution on the performed action given the observed descriptor, $p(a_i \mid f_t^k)$, $a_i \in A$, $\sum_{i=1}^{N} p\left(a_i \mid f_t^k\right) = 1$. This posterior probability distribution measures the uncertainty of the observed descriptors of being an instance of each one of the categories.

The posterior probability distributions computed for each one of the K views of the scene are combined using a classifier fusion algorithm, generating a posterior distribution $p\left(\alpha_t \mid f_t^1 \ldots f_t^K\right)$ on the action performed given the descriptor computed by the different views.

Finally, the posterior probability distributions created at each instant t are entered into a sequence classifier to generate a single posterior distribution on the performed action given the observation sequence $p\left(\alpha \mid f_1^1 \ldots f_T^K\right)$. This distribution will be finally used to predict the action of the observed individual.

This architecture distributes the decision making process across multiple nodes, following the DEI-DEO data fusion paradigm shown on chapter 3. Each node processes the image grabbed from each camera, and makes a partial decision on the action using the information contained just in that image. A central node then grabs the decisions taken by each node and combines them to make the final decision on the performed action. One advantage of this approach is that if a camera breaks the action recognition decision can still be made, as the central node would be still collecting the decisions made by the other nodes. Other advantage is that the computational resources needed to process the image sequences are allocated across different nodes, reducing the amount of resources needed at the central node.

A possible alternative way of structuring the system would be to first classify each sequence at each camera and then sending just one posterior distribution to the central node, as in (Srivastava *et al.* , 2009). However, we are interested in performing frame by frame action segmentation at the central node in the future, assuming different actions happen on the input sequences. If the system would be structured in such way it would be more difficult to make this extension.

6.2 Single view processing

6.2.1 Human action representation

The first step in the proposed architecture is to compute a descriptor to capture the variability of the actions to predict. The motion descriptor employed in this systems is the proposed in (Tran & Sorokin, 2008) shown on appendix C.1 and already employed in previous chapter.

6.2.2 Dimensionality reduction

The action descriptor employed has a large dimensionality ($D_{TRAN} = 286$), and needs to be projected into a lower dimensional space in order to prevent the problems derived from "the curse of dimensionality". Any method in the graph embedding framework might be employed for this task. Here PCA (see chapter 4) is employed. Previous chapter only focused on unsupervised dimensionality reduction methods, i.e., methods not employing class information to compute the low dimensional representation of the data. Supervised dimensionality reduction methods might be also understood as instantiations of the graph embedding framework. Linear Discriminant Analysis is the supervised counterpart of PCA and is going to be also employed. Detail on the formulation of LDA in the graph embedding framework might be found on (Yan *et al.* , 2007).

6.2.3 Action classification

The action descriptor f_t^k computed at each frame is introduced into a probabilistic classifier in order to generate the posterior probabilities of the performed action given the evidence grabbed at that instant. A parametric (k-means + naive Bayes) and a non-parametric (Nearest neighbor conditional density estimator) density estimators are going to be employed to test the proposed system. The parametric splits the

feature space in different regions, estimating the conditional probabilities of each class at each region. The non-parametric estimates the conditional probabilities of each class according to the neighbourhood of a test point. This way the possibility of using a local or a global approaches to classification is incorporated to the system.

6.2.3.1 Nearest neighbor conditional density estimator

The nearest neighbor conditional density estimator (kNN) (Bishop *et al.* , 2006) is a well-known non-parametric conditional density estimator. The estimator locally captures the conditional density around a given test point x. Let K be a fixed neighborhood size and K_i, $\sum_i K_i = K$ the number of neighbors of class a_i

$$p\left(x \mid a_i\right) = \frac{K_i}{K} \tag{6.1}$$

6.2.3.2 K-means + naive Bayes

The space of feature descriptors f_t^c will be quantified using a codebook of size K. Each feature vector will be associated with its nearest center to obtain the word w_k. Codebook centers are computed using the k-means algorithm.

$$p\left(w_k \mid a_i\right) = \frac{p\left(a_i \mid w_k\right) p\left(w_k\right)}{p\left(a_i\right)} \tag{6.2}$$

6.3 Action fusion

After extracting a set of posterior probability distributions $p(a_t^k \mid f_t^k)$ from the frame descriptor f_t^k computed for each view, they have to be combined to generate a joint posterior probability distribution $p(\alpha_t \mid f_t^1, \ldots, f_t^K)$ representing the uncertainty in the classification with respect to the evidence perceived by the different cameras at an instant t.

Two different algorithms will be tested for this task. The first is a voting scheme. The second is a Bayesian network modeling the errors in local classifications.

6.3.1 Voting

The first algorithm that we tested for the fusion of single view soft classifications is defined as the product of the posterior probabilities.

$$p\left(\alpha_t \mid f_t^1, \ldots, f_t^K\right) \propto \prod_{k=1}^{K} p\left(a_k \mid f_t^k\right) \tag{6.3}$$

This algorithm is tested as baseline to measure the efficiency of the bayesian network.

6.3.2 Bayesian network

The second algorithm that we tested for the fusion of single view soft classifications is based on the Bayesian network shown in Figure 6.2. The network is composed of observation nodes f_t^k, representing the observation at instant t and camera k, a node α_t representing the activity at time t and a set of latent nodes a_t^k to model the single view classification.

Given a set of frame descriptors $\mathbf{f}_t = f_t^1, \ldots, f_t^K$, a set of latent variables $\mathbf{a}_t = a_t^1, \ldots, a_t^K$, and the activity label α_t, their joint probability is factorized as

$$P\left(\alpha_t, \mathbf{a}_t, \mathbf{f}_t\right) = P\left(\alpha_t \mid \mathbf{a_t}\right) P(\mathbf{a_t} \mid \mathbf{f_t}) P(\mathbf{f_t}) \tag{6.4}$$

.

The conditional probability given $\mathbf{f_t}$ is then:

$$P\left(\alpha_t, \mathbf{a}_t \mid \mathbf{f}_t\right) = \frac{P\left(\alpha_t, \mathbf{a}_t, \mathbf{f}_t\right)}{P(\mathbf{f}_t)} = P\left(\alpha_t \mid \mathbf{a_t}\right) P(\mathbf{a_t} \mid \mathbf{f_t}) \tag{6.5}$$

.

The probability $P(\alpha_t, \mathbf{a}_t, \mathbf{f}_t)$ is defined as a product of independent factors, assuming hidden variables a_t^c to be independent:

$$P(\alpha_t \mid \mathbf{a_t}) \doteq \prod_{k=1}^{K} P\left(\alpha_t \mid a_t^k\right) \tag{6.6}$$

.

With this assumption we rule out modeling correlations between local classification errors. In this way, this assumption reduces to two the exponential number of probability distributions that would otherwise need to be estimated. Thus, equation 6.5 can be rewritten as

$$P(\alpha_t, \mathbf{a}_t \mid \mathbf{f}_t) = \prod_{k=1}^{K} p\left(\alpha_t \mid a_t^k\right) p\left(a_t^k \mid f_t^k\right) \tag{6.7}$$

.

Marginalizing over a_t^k:

$$P(\alpha_t \mid \mathbf{f}_t) = \prod_{k=1}^{K} \sum_{a^k} p\left(\alpha_t \mid a_t^k\right) p\left(a_t^k\right) p\left(f_t^k \mid a_t^k\right) \tag{6.8}$$

.

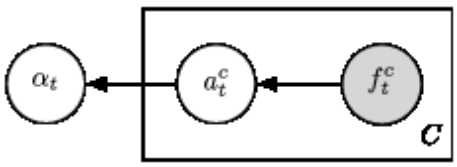


Figure 6.2: Plate model of the Bayesian network used to combine the outputs from the classifiers at each camera

Bayesian network parameters are estimated using labeled training samples. $p\left(a_t^k \mid f_t^k\right)$ is known, being provided by the single view soft classifiers, so only $p(\alpha^t \mid a_k^t)$ needs to be estimated. Let $O^k = \left(o_1^k, \ldots, o_L^k\right\}$ be the set of L training frame descriptors computed at camera c with their respective activity labels $Y^c = \{y_1^c, \ldots, y_L^c\}$, $y_l^c \in A$.

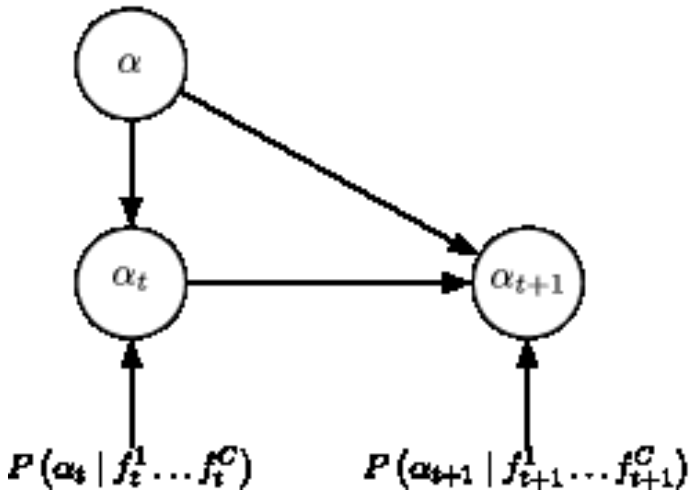


Figure 6.3: Dynamic Bayesian network for sequence classification

Model parameters are estimated as

$$p\left(\alpha^t = a_i \mid a_t^c = a_j\right) = \frac{\sum_{l=1}^{L} \gamma_l p\left(a_t^k = a_j \mid o_l^K\right)}{\sum_{n=1}^{N}\sum_{l=1}^{L} \gamma_l p\left(a_t^k = a_n \mid o_l^k\right)} \tag{6.9}$$

where $\gamma_l = 1$ if $y_l = a_j$ and $\gamma_l = 0$ otherwise.

6.4 Sequence classification

Human actions are not isolated occurrences, they happen in sequence. By this time, the reader will probably have noted the t subscript in our formulation. The method proposed until now considers individual frame descriptors, but ignores sequence dynamics. So, given a sequence of frame descriptors computed at each camera $F = \{f_1^1, \dots, f_1^K, \dots, f_T^1, \dots, f_T^K\}$, we need to associate it with their respective activity α, assuming that there is only one activity performed in the sequence. The sequence length T is not needed to be the same for all sequences.

In this paper a discriminative Hidden Markov Model (HMM) (Rabiner, 1989)

is employed for this task. The probability of a path of hidden node values $H = a_1, \ldots, a_T$ given an action class a and an observed sequence F is defined as

$$\begin{aligned} p(H \mid F, a) = & p(a_1 \mid a)\, p\left(a_1 \mid f_1^1 \ldots f_1^K\right) \\ & \prod_{t=2}^{T} p(a_t \mid a_{t-1}, a)\, p\left(a_t \mid f_t^1 \ldots f_t^K\right) \end{aligned} \tag{6.10}$$

where $p(a_t \mid a_{t-1}, a)$ is a transition model for each action. This factorization of the probability distribution is graphically shown on figure 6.3. The action a^* performed given a sequence of observed actions F is

$$a^* = \arg\max_a p(a \mid F) \tag{6.11}$$

where $p(a \mid F)$ is defined as

$$p(a \mid F) \propto \sum_{a_T} p(a_T \mid F, a)\, p(a) \tag{6.12}$$

The above quantity can be recursively estimated using the standard forward-backward procedure (Rabiner, 1989).

The parameters of the model, $p(a_1 \mid a)$ and $p(a_t \mid a_{t-1}, a)$, can be estimated from labeled training samples in a similar way as for the Bayesian network in section 6.3.2. We assume a uniform prior on $p(a)$.

6.5 Experiments

6.5.1 Experimental setup

The performance of the proposed systems is going to be evaluating employed IXMAS dataset (see section B.2 for additional details), employing the LOAO-CV evaluation protocol as was done to test the system in previous chapter.

PCA and LDA are used to project the frame descriptors into a lower dimensional subspace. In the case of PCA, it has been tested for $d = \{10, 15, 20, 25\}$.

The size of the codebook used in the BN classifier has been experimentally adjusted to $k = 300$ words. The k-NN density estimators will be tested using $k = 3$, $k = 5$ and $k = 7$ neighbors.

A different dimensionality reducer and classifier is trained for each camera in the system, with the images they grabbed. Classifier fusion and sequence classifiers are then run on the results provided by these classifiers.

6.5.2 Results

6.5.2.1 Single camera classification

Table 6.1 shows the accuracy of the single frame classifiers. Irrespective of the frame descriptors used, the results reported for cameras $1 - 4$ are quite similar, whereas the accuracy drops by around 10% for camera 5.

PCA projections seems to have a better performance than LDA when the number of dimensions is high enough. Regarding the classification algorithms employed, k-NN algorithms are more accurate than the BN algorithm for almost all the choices of projection algorithm. As regards the choice of the number of neighbors to use, 7-NN was found to return better results than 3-NN and 5-NN, but the difference is not substantial.

6.5.2.2 Classifier fusion

Table 6.2 shows the accuracies achieved after applying the classifier fusion algorithms to the posterior distribution generated from each camera. We find that whereas the voting algorithm always improves the accuracy of the NB classifiers at least a little, this is not the case for the k-NN classifiers, where the final accuracy is always worse

		Classifier			
Camera	Reducer	NB	3-NN	5-NN	7-NN
	LDA	0.4740	0.4650	0.4878	0.4952
	PCA_{10}	0.4292	0.4295	0.4495	0.4611
	PCA_{15}	0.4355	0.4736	0.4908	0.4970
	PCA_{20}	0.4432	0.4864	0.5045	0.5123
1	PCA_{25}	0.4549	0.5064	0.5179	0.5218
	LDA	0.4908	0.4848	0.5072	0.5162
	PCA_{10}	0.4108	0.4161	0.4360	0.4415
	PCA_{15}	0.4249	0.4564	0.4742	0.4784
	PCA_{20}	0.4440	0.4797	0.5001	0.5012
2	PCA_{25}	0.4534	0.4948	0.5095	0.5145
	LDA	0.4732	0.4652	0.4866	0.4969
	PCA_{10}	0.4241	0.4508	0.4653	0.4707
	PCA_{15}	0.4660	0.5049	0.5214	0.5249
	PCA_{20}	0.4693	0.5236	0.5416	0.5453
3	PCA_{25}	0.4779	0.5283	0.5447	0.5501
	LDA	0.5084	0.5066	0.5282	0.5390
	PCA_{10}	0.4315	0.4341	0.4541	0.4638
	PCA_{15}	0.4487	0.4759	0.4911	0.4966
	PCA_{20}	0.4650	0.4976	0.5131	0.5199
4	PCA_{25}	0.4799	0.5227	0.5389	0.5444
	LDA	0.3407	0.3209	0.3517	0.3604
	PCA_{10}	0.3656	0.4005	0.4244	0.4373
	PCA_{15}	0.3652	0.4231	0.4458	0.4545
	PCA_{20}	0.3710	0.4348	0.4536	0.4568
5	PCA_{25}	0.3650	0.4444	0.4562	0.4568

Table 6.1: Results obtained after single camera classification of the IXMAS dataset

than for the best single view classifier. However, the accuracy provided by the BN algorithm is always better than the best single view classifier by about $10\% - 20\%$.

6.5.2.3 Sequence classification

Finally, the results for sequence classification are shown in table 6.3. The accuracy improvement is notable when compared with frame-by-frame classification. Confusion matrix of the best classifier found is shown on figure 6.4

The behavior of the sequence classification algorithm depends on the origin of the instant classification posteriors that it combines. When using the output from the NB classifier fusion algorithm, the result varies slightly with respect to the number of dimensions used in the frame descriptor for any given classifier. In the case of k-NN classifiers some overfitting can be observed, as the final accuracy starts to drop as the dimensionality grows. When using the voting algorithm, the variation of the results

		Classifier			
Fusion method	Reducer	NB	3-NN	5-NN	7-NN
Vote	LDA	0.6193	0.4121	0.4769	0.5140
	PCA_{10}	0.5603	0.4400	0.4942	0.5178
	PCA_{15}	0.5773	0.4775	0.5246	0.5507
	PCA_{20}	0.5915	0.4950	0.5390	0.5674
	PCA_{25}	0.5932	0.5042	0.5512	0.5713
Bayesian Network	LDA	0.6198	0.6310	0.6475	0.6523
	PCA_{10}	0.5501	0.5894	0.6055	0.6111
	PCA_{15}	0.5712	0.6292	0.6404	0.6450
	PCA_{20}	0.5834	0.6505	0.6609	0.6629
	PCA_{25}	0.5854	0.6601	0.6679	0.6698

Table 6.2: Accuracy obtained after applying classifier fusion algorithms to the IXMAS dataset

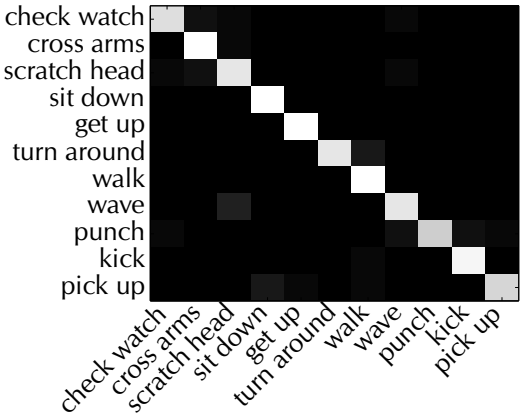


Figure 6.4: Confusion matrix for the best system configuration found

		Classifier			
Fusion Method	Reducer	NB	3-NN	5-NN	7-NN
Vote	LDA	0.8649	0.8559	0.8649	0.8589
	PCA_{10}	0.8018	0.8348	0.8438	0.8228
	PCA_{15}	0.8348	0.8468	0.8709	0.8649
	PCA_{20}	0.8438	0.8619	0.8799	0.8859
	PCA_{25}	0.8529	0.8709	0.8769	0.8709
Bayesian Network	LDA	0.8348	0.9009	0.8979	0.8979
	PCA_{10}	0.5075	0.8589	0.8468	0.8348
	PCA_{15}	0.5886	0.9009	0.8949	0.8859
	PCA_{20}	0.6096	0.9159	0.9069	0.9009
	PCA_{25}	0.6246	0.8919	0.9009	0.8979

Table 6.3: Accuracy obtained after applying the sequence classification algorithm to the IXMAS dataset

is greater. While the behavior is similar to BN's when applied to Tran's descriptor, the result quickly overfits when applied to the MHI descriptor and drops with the dimensionality.

6.5.3 Discussion

The BN classifier fusion algorithm has been proved to outperform the voting algorithm. The reason is that the BN attaches different weights to the posteriors produced by each camera, according to a model of the usual errors in the classification, whereas the voting algorithm does not use any prior information about classification accuracy.

The results for sequence classification, when compared to instant classification, show that actions are not isolated occurrences, but happen in sequence. It is not enough to consider just one instant in order to recognize actions, and, whenever already available, the past and the future frames have to be employed to make the decision about what is happening or happened.

When globally examining the results, there is one discouraging observation: the best algorithm configuration found for one tier of the system does not guarantee that the best accuracy will be achieved on the next tier up. We observed many times that the accuracy given after the classifier fusion by the classifiers with the best single frame performance is smaller than the reported for other classifiers with a worse performance at the single frame level. There are also similar examples of these phenomena involving the classifier fusion and sequence classification results. This implies that action recognition systems cannot be constructed incrementally in order to find the best configuration, as the configuration with the best result at the highest level is not the configuration with the best result at intermediate levels.

The accuracy of the proposed system is compared to other proposals reporting results on the IXMAS dataset. Table 6.4 compares the proposed system to other alternatives. All algorithms are deterministic for a fixed training set. To the best

Method	Accuracy	Type
(Tran & Sorokin, 2008)	81	2D
(Srivastava *et al.* , 2009)	81.4	Multicamera
Chapter 5	91.26	Multicamera
Proposed	91.59	Multicamera
(Weinland *et al.* , 2006)	93.33	3D
(Peng *et al.* , 2009)	94.59	3D

Table 6.4: Comparison of the accuracy of the proposed method to other works evaluated with IXMAS dataset

of our knowledge, the proposed system achieves an accuracy similar to the best reported to date (Peng *et al.* , 2009). Let us stress that while the best result was based on the classifications of the 3D visual hull, this proposal relies on only well-known simple 2D pattern recognition techniques, without any need of recovering camera calibration parameters.

Finally, we want to point out that the accuracy of the system proposed here is almost the same than the reported for the system on chapter 5. This is quite surprising because the philosophies that have ruled the development of both systems are completely different. Proposal on previous chapter models the correlations among the observation extracted from each camera, while the proposed here is based on modelling the failure in the prediction of the different actions produced by the different cameras. A look into the reported confusion matrices points out that the main errors are not exactly the same, so both approaches might have a complementary behavior that might be exploited to create new systems.

6.6 Remarks

This paper has presented a distributed human action recognition system. 2D descriptors have been extracted for the frames captured at each one of the available views. They have been projected into a lower dimensional space and introduced into a probabilistic classifier to generate a posterior probability of the performed ac-

tion. The posteriors for the different cameras have been merged using a classifier fusion algorithm, whose results have been fed into a sequence classifier to make the final decision on the performed action. The system has been tested with different algorithms, exploiting the flexibility provided by the well-defined interfaces between levels. As result, the system achieves an accuracy similar to the state-of-the-art of human action recognition algorithms for classifying the IXMAS dataset.

7

Conclusions

I'm Mr. Wolf. I solve problems

Pulp Fiction

Human Action Recognition is one of the main topics of current research by the computer vision community. The arise of VSNs has impossed new restrictions to the complexity of the algoritms employed for the task. Computational complexity, bandwidth usage and energy consuption have to be minimized in order to deploy human action recognition systems with VSNs.

The main objective of this thesis was to design new algorithms taking into account these design constraints and, in the way to fulfill these requirements different general approaches for solving problems have been designed.

The main contributions of this thesis might be summarized as follows:

- A procedure to train HCRF sequence classifiers performing model and feature selection. Employing the right features and the proper model complexity not only reduces the computational load compared to the standard model, it also increases the predictive accuracy of the models. The Occam's Razor principle of machine learning has shown to be true.

- A multiple view dimensionality reduction framework has been proposed as an extension to the graph embedding framework. The framework abstract the

formulation of existing multiple view dimensionality reduction algorithms and allows the formulation of new ones. The usage of approximate methods allows obtaining solutions for the framework in large scale learning scenarios. The framework has been validated in a multicamera human action recognition task, achieving state of the art results.

- An alternative proposal to perform human action recognition from multiple cameras has been developed bringing action prediction to the camera nodes. A probabilistic formulation allows the combination of the decision made at the camera nodes to make a global one. This method has also shown results similar to the achieved by state of the art proposals.

This dissertation has shown that accurate multicamera human action recognition methods might be designed without the need of performing the 3D reconstruction of the scene at a central node. Bringing processing to the camera nodes reduces the amount of data that should be streamed over the network, allowing the usage of human action recognition methods in resource-constrained environments.

This dissertation has also provided a review of the state of the art methods for the recognition of human actions. The different steps that should be performed in order to bridge the semantic gap between pixel intensity values and descriptions have been discussed. The recognition of human actions from multiple cameras has been analyzed from the view points of data fusion systems. Dasarathy's Input-Output model has shown to be an effective framework to categorize existing works.

7.1 Future Work

Future research lines that arise from the work presented here are going to be discussed now.

7.1.1 New instantiations of the multiple view dimensionality reduction framework

This dissertation has only proposed one new algorithm employing the formalism of the multiple view graph embedding framework, extending the Isomap algorithm. This extension has been made with the nature of the motion data to be analyzed in mind. However, different single view graph embedding algorithms might be extended to the multiple view case. Extensions of Laplacian Eigenmaps or Locally Linear Embedding might be proposed. These extensions would lead to multiple dimensionality reduction algorithm to analyze data structured in high dimensional clusters, instead of data with a continuous distribution in the high dimensional space.

7.1.2 Beyond Model and Feature selection for the HCRF

This thesis has proposed a procedure to train HCRF sequence classifiers incorporating model and feature selection. HCRFs belong to the general class of log-linear models. Other models in this family also incorporate the usage of hidden variables, such the Latent-Dynamic Hidden Conditional Random Field (LD-HCRF) (Morency *et al.* , 2007) or the more general Dynamic Conditional Random Fields (DCRFs) (Sutton *et al.* , 2007). The procedure for model selection presented in this thesis might be applied also for these models, defining appropriate group structures. In particular, the usage of LD-HCRFs allows to perform sequence segmentation tasks, a problem more challenging than just sequence classification with a great importance for the recognition of human actions in real time.

This thesis has employed batch optimization algorithms, where the optimization direction is computed employing all the training samples. In large scale learning scenarios, online optimization algorithms, where the direction is computed with an small subset of the training samples or even a single one, has shown a faster convergence rate. However, to the best of our knowledge, an online optimization algorithm for parameter estimation of overlapping group-L1 regularized log-linear

models has not been proposed yet. Defining such an algorithm will be a challenging task.

7.1.3 Multicamera Human Action Recognition with sparse coding

Sparse coding methods (Lee *et al.* , 2007) have received an increasing attention in recent works, with application to a wide variety of computer vision tasks. Sparse coding algorithms reconstruct a signal employing only a few atoms of an overcomplete dictionary. Multiple algorithms have been proposed to obtain the optimal reconstruction coefficients and to learn optimal overcomplete dictionaries for signal reconstruction.

A feature fusion method as the proposed in this thesis might be built employing sparse coding. The algorithms to obtain the optimal reconstruction coefficients have to be extended to account for the multiple views of the data. The existing algorithm to learn the overcomplete dictionaries probably are already prepared to deal with the multiple view case.

7.1.4 View-Invariant action recognition

With the multiple-view learning framework introduced in this thesis a subspace has been learned representing the information shared by the multiple cameras observing the scene. With the spectral regression framework transformation from the high dimensional space to the low dimensional have been learned, one for each camera. The low dimensional representations obtained for the different camera viewpoints in this way are similar. What if a single function is learned, instead of one for each camera? What if that function is employed to project action sequences captured from camera viewpoints not known during training? Would it project them to points similar to the known cameras? How many viewpoints are needed to estimate this hypothetical function? It is not clear if that function exists, but if it does, it would be a

way to achieve viewpoint invariance in the prediction of human actions. Viewpoint invariance allows fast deployment of human action recognition systems, as they don't have to be trained from the viewpoint that they will employ.

A

Published Results

THIS appendix lists the specific works that have to do with this thesis:

- R. Cilla, M.A. Patricio, A. Berlanga, and J.M. Molina. A probabilistic, discriminative and distributed system for the recognition of human actions from multiple views. Neurocomputing, 2011.

- R. Cilla, M. Patricio, A. Berlanga, and J. Molina. On the process of designing an activity recognition system using symbolic and subsymbolic techniques. International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008), pages 729–738, 2009.

- R. Cilla, M. Patricio, A. Berlanga, and J. Molina. Fusion of single view soft k-nn classifiers for multicamera human action recognition. Hybrid Artificial Intelligence Systems, pages 436-443, 2010.

- R. Cilla, M. Patricio, A. Berlanga, and J. Molina. Evaluating manifold learning methods and discriminative sequence classifiers in view-invariant action recognition. User-Centric Technologies and Applications, pages 11–18, 2011.

- R. Cilla, M. Patricio, A. Berlanga, and J. Molina. Improving the accuracy of action classification using view-dependent context information. Hybrid Artificial Intelligent Systems, pages 136–143, 2011.

- R. Cilla, M. Patricio, A. Berlanga, and J. Molina. Multicamera action recognition with canonical correlation analysis and discriminative sequence classification. In 4th International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2011, pages 491–500. Springer, June 2011.

- R. Cilla, M.A. Patricio, A. Belanga, and J.M. Molina. Non-supervised discovering of user activities in visual sensor networks for ambient intelligence applications. In Applied Sciences in Biomedical and Communication Technologies, 2009. ISABEL 2009. 2nd International Symposium on, pages 1–6. IEEE, 2009.

- R. Cilla, M.A. Patricio, A. Berlanga, and J.M. Molina. Creating human activity recognition systems using pareto-based multiobjective optimization. Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on, pages 37–42. IEEE, 2009.

- R. Cilla, M.A. Patricio, A. Berlanga, and J.M. Molina. Phd forum: Non supervised learning of human activities in visual sensor networks. In Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on, pages 1–2. IEEE, 2009.

B

Datasets Employed

THIS appendix shows the datasets employed to validate the proposals introduced in this dissertation.

B.1 Weizmann

Weizmann dataset contains 90 low-resolution video sequences showing 9 different people performing 10 different actions. Images are recorded at 50 fps with a resolution of 180 x 144 pixels. Weizmann dataset is one of the most simple testbeds to evaluate single camera human action recognition methods. Many works have reported perfect prediction for them. The actions contained in the video are: (1) *run*, (2) *walk*, (3) *skip*, (4) *jumping-jack* (or shortly *jack*), (5) *jump-forward-on-two-legs* (or *jump*), (6) *jump-in-place-on-two-legs* (or *pjump*), (7) *gallopsideways* (or *side*), (8) *wave-two-hands* (or *wave2*), (9)*wave-one-hand* (or *wave1*) and (10) *bend*. Sample frames of these actions are shown on figure B.1.

B.2 Ixmas

IXMAS (Inria Xmas Motion Acquisition Sequences) dataset (Weinland *et al.* , 2006) contains 13 actions performed by 12 different actors at least 3 times. The action

sequences are simultaneously recorded from 5 different viewpoints at 23 fps with a resolution of 390x291 pixels. IXMAS dataset is the standard testbed to measure the performance of multicamera human action recognition methods and viewpoint invariant action recognition methods. Experiments reported in the literature have limited their predictions to a subset of 11 actions employing only 10 actors. These actions are: (1) *check watch*, (2) *cross arms*, (3) *scratch head*, (4) *sit down*, (5) *get up*, (6) *turn arround*, (7) *walk*, (8) *wave*, (9) *punch*, (10) *kick* and (11) *pick up*. A sample frame of the *kick* action observed by the 5 viewpoints is presented in figure B.2

(a) Bend (b) Jack (c) Jump

(d) PJump (e) Run (f) Side

(g) Skip (h) Walk (i) Wave1

(j) Wave2

Figure B.1: Sample frames from Weizmann dataset

(a) Cam 0

(b) Cam 1

(c) Cam 2

(d) Cam 3

(e) Cam 4

Figure B.2: A frame belonging to the action *kick* of IXMAS dataset seen from the 5 available views

C

Feature Extraction

THIS appendix presents the motion descriptors employed in the different experiments that has been shown in this thesis.

C.1 Tran's descriptor

The actor descriptor proposed in (Tran & Sorokin, 2008) combines optical flow and appearance information. It has been choosen to be employed in the systems validated with Ixmas dataset because it has shown a high experimental performance on single camera applications.

To compute it, the bounding box of the human being is normalized to a square box preserving aspect ratio. Shape and optical flow are extracted from the box. Vertical and horizontal planes of the optical flow are split and blurred with a median filter. Then, each box has three channels: silhouette, vertical flow and horizontal flow. The box is divided into 4 tiles, and a radial 18-bin histogram is computed from each tile and each channel. The obtained histograms are concatenated to obtain a 216-d vector. Lastly, PCA reduction of the surrounding past, present and future vectors is appended to finally generate a descriptor of $d_{TRAN} = 286$ dimensions. Readers are referred to (Tran & Sorokin, 2008) for more details.

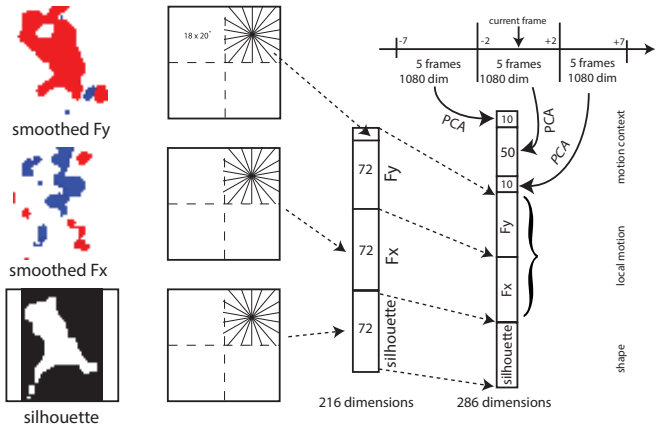


Figure C.1: Tran's descriptor, reproduced from (Tran & Sorokin, 2008)

C.2 Euclidean Distance Transform

The distance transform of an object replaces each pixel by the sortest distance to the outside of the object. Be S the set of pixels of a given binary image I. The distance transform is formally defined for every pixel $s \in S$ as:

$$DT(s) = \begin{cases} \min_q \|s - q\| : I(q) = 0, \quad q \in S & \text{if } I(s) = 1 \\ 0 & \text{if } I(s) = 0 \end{cases} \tag{C.1}$$

The distance transform is computed for the silhouettes of the Weizmann dataset. Silhouettes are rescaled to a box of 64x48 pixels. Distance transform is computed for each pixel. Pixel values are concatenated to form a descriptor with dimensionality $d_{DT} = 3072$.

D

Dynamic Time Warping Nearest Neighbor Sequence Classification

GIVEN two temporal sequences $X = x_1, x_2, \dots, x_N$ and $Y = y_1, y_2, \dots, y_M$ of respective lengths N and M, the Dynamic Time Warping (DTW) distance between the sequences, denoted by $d_{DTW}(X, Y)$ measures the cost of transforming one qsequence into the other in the sense of the number of movements, insertions and deletions required for the transformation.

Given a dataset of $D = \left\{X^i, Y\left(X^i\right)\right\}$, $1 \leq i \leq N$ of sequences X^i with their corresponding labels $Y\left(X^i\right) \in \{y_1, \dots, y_C\}$, a nearest neighbor classifier might be build employing the DTW distance. Given a test sequence X', the set $\mathcal{N}_K(Y)$ denotes the K sequences in the dataset D closer to X' in the sense of the nearest neighbor distance. The nearest neighbor score of each class is given by:

$$NN_c\left(X'\right) = \sum_{X \in \mathcal{N}_K(Y)} \frac{\delta\left(c, Y\left(X\right)\right)}{d_{dtw}\left(X, X'\right)^2} \tag{D.1}$$

wher the function $\delta(a, b) = 1$ if $a = b$ and $\delta(a, b) = 0$ otherwise. The class of a test sequence X' is that maximizing the nearest neighbor score:

$$Y\left(X'\right) = \arg\max_c NN_c\left(X'\right) \tag{D.2}$$

E

Evaluation Protocols and Metrics

THIS appendix presents the evaluation protocol employed for the validation of the different proposed methods and the metrics to be employed for their quantitative comparison.

E.1 Leave One Actor Out Cross-Validation

The evaluation protocol employed to measure the performance of the proposed methods is Leave One Actor Out - Cross-Validation (LOAO-CV). Examples in the dataset are split in different folds according to the actor performing the actions. Models are trained leaving one fold out, employed for validation. This process is repeated until the folds from every actor have been employed once for system validation.

E.2 Metrics

Two different measures are going to be employed to asses the performance of the proposed methods:

- **Recognition Rate**. It is defined as the number of correctly classified samples divided by the number of total samples. It is commonly employed in pattern

recognition applications to measure the experimental performance. It is defined as:

$$\text{Recognition Rate} = \frac{\text{\# of correct predictions}}{\text{\# of samples}} \tag{E.1}$$

- **Negative Log-likelihood**. Given a probabilistic model λ and a sample x, it is defined as:

$$\text{nll} = -\log P(x \mid \lambda) \tag{E.2}$$

This measure is employed to show that the HCRF learned with the proposed regularization strategy predicts better unknown samples than those trained with the standard procedure.

References

Achard, Catherine, Qu, Xingtai, Mokhber, Arash, & Milgram, Maurice. 2007. A novel approach for recognition of human actions with semi-global features. *Machine Vision and Applications*, **19**(1), 27–34. 2.4.2.1

Aggarwal, J.K., & Cai, Q. 1999. Human motion analysis: a review. *Pages 90–102 of: Proceedings IEEE Nonrigid and Articulated Motion Workshop*, vol. 73. IEEE Comput. Soc. 2.2

Aggarwal, J.K., & Ryoo, M.S. 2011. Human activity analysis. *ACM Computing Surveys*, **43**(3), 1–43. 1, 2.2, **??**, 2.3, 2.3

Ali, Saad, Member, Student, & Shah, Mubarak. 2010. Human Action Recognition in Videos Using Kinematic Features and Multiple Instance Learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **32**(2), 288–303. 2.4.2.2

Azary, Sherif, & Savakis, Andreas. 2010. View Invariant Activity Recognition with Manifold Learning. 606–615. 2.4.4

Bach, F.R., & Jordan, M.I. 2003. Kernel independent component analysis. *The Journal of Machine Learning Research*, **3**, 1–48. 5.2.4

Baker, S., Scharstein, D., Lewis, JP, Roth, S., Black, M.J., & Szeliski, R. 2011. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, **92**(1), 1–31. 2.4.2.2

Bauschke, H.H., & Lewis, A.S. 2000. Dykstras algorithm with bregman projections: A convergence proof. *Optimization*, **48**(4), 409–427. 4.2.1

Belkin, M., & Niyogi, P. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems*, **14**, 585–591. 2.4.4

Bishop, C.M., *et al.* . 2006. *Pattern recognition and machine learning*. Vol. 4. Springer New York. 4.1, 6.2.3.1

Blackburn, Jaron, & Ribeiro, Eraldo. 2007. Human Motion Recognition Using Isomap and Dynamic Time Warping. *Proceedings of the 2nd conference on Human motion: understanding, modeling, capture and animation*. 2.5.1.1

Bobick, Aaron F. 1997. Movement, activity and action: the role of knowledge in the perception of motion. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, **352**(1358), 1257–65. **??**, 2.3, 2.3

Bobick, Aaron F., & Davis, James W. 2001. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(3), 257–267. 2.4.2.1

Boyd, S., & Vandenberghe, L. 2004. *Convex optimization*. Cambridge University Press. 4.1.1, 4.2.1, 4.2.1

Brand, Matthew, & Kettnaker, Vera. 2000. Discovery and Segmentation of Activities in Video. **22**(8), 844–851. 2.5.1.2

Bregonzio, M. 2009. Recognising action as clouds of space-time interest points. *Pages 1948–1955 of: 2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2.4.3

Bremond, F., & Nevatia, Ramakant. 2000. Representation and optimal recognition of human activities. *Pages 818–825 of: Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, vol. 1. IEEE Comput. Soc. **??**, 2.3

Cai, D., He, X., & Han, J. 2007a. Isometric projection. *Page 528 of: Proceedings of the National Conference on Artificial Intelligence*, vol. 22. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999. 2.4.4, 5.1.5

Cai, D., He, X., & Han, J. 2007b. Spectral regression for efficient regularized subspace learning. *Pages 1–8 of: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE. 5.1.1, 5.3.1

Cédras, Claudette, & Shah, Mubarak. 1995. Motion-based recognition a survey. *Image and Vision Computing*, **13**(2), 129–155. 2.2

Charfi, Y., Wakamiya, N., & Murata, M. 2009. Challenging issues in visual sensor networks. *IEEE Wireless Communications*, **16**(2), 44–49. 1.1

Chaudhry, R., Ravichandran, A., Hager, G., & Vidal, R. 2009. Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. *Pages 1932–1939 of: 2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2.4.2.2, 2.4.4, 2.5.1.2, 6.1

Chen, Daniel, Chou, PC, & Fookes, CB. 2008. Multi-view human pose estimation using modified five-point skeleton model. 17–19. 3.3.2

Chomat, Olivier, & Crowley, James L. 2000. A Probabilistic Sensor for the Perception and the Recognition of Activities. 487–503. 2.4.2.2

Corradini, Andrea. Dynamic time warping for off-line recognition of a small gesture vocabulary. *Proceedings IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, 82–89. 2.5.1.1

Cutler, Ross, & Turk, Matthew. 1998. View-based Interpretation of Real-time Optical Flow for Gesture Recognition. *Pages 416–421 of: Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*. 2.4.2.2

Dasarathy, B.V. 1997. Sensor fusion potential exploitation-innovative architectures and illustrative applications. *Proceedings of the IEEE*, **85**(1), 24–38. 3.2.2

Davenport, M.A., Hegde, C., Duarte, M.F., & Baraniuk, R.G. 2010. Joint manifolds for data fusion. *Image Processing, IEEE Transactions on*, **19**(10), 2580–2594. 5.2.5

Dollar, Piotr, Rabaud, V., Cottrell, Garrison, & Belongie, Serge. 2005. Behavior Recognition via Sparse Spatio-Temporal Features. *Pages 65–72 of: 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. IEEE. 2.4.3

Efros, Alexei A, Berg, Alexander C, Mori, Greg, Malik, Jitendra, & Division, Computer Science. 2003. Recognizing action at a distance. *Pages 726–733 vol.2 of: Proceedings Ninth IEEE International Conference on Computer Vision*. IEEE. 2.4.2.2, 2.5.1.1

Escobar, Maria-Jose, Masson, Guillaume S., Vieville, Thierry, & Kornprobst, Pierre. 2009. Action Recognition Using a Bio-Inspired Feedforward Spiking Network. *International Journal of Computer Vision*, **82**(3), 284–301. 2.4.2.1

Fanti, Claudio, Zelnik-manor, Lihi, & Perona, Pietro. 2005. Hybrid Models for Human Motion Recognition. *Pages 1166—-1173 of: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 00. 2.4.1

Gavrila, D M, & Davis, L S. 1995. Towards 3-D model-based tracking and recognition of human movement: a multi-view approach. 3–8. 2.4.1

Gavrila, D.M. 1999. The visual analysis of human movement: A survey. *Computer vision and image understanding*, **73**(1), 82–98. 2.2

Gkalelis, Nikolaos, Kim, Hansung, Hilton, Adrian, Nikolaidis, Nikos, & Pitas, Ioannis. 2009. The i3DPost Multi-View and 3D Human Action/Interaction Database. *2009 Conference for Visual Media Production*, Nov., 159–168. 3.3.2

Gorelick, Lena, Blank, Moshe, Shechtman, Eli, Irani, Michal, & Basri, Ronen. 2007. Actions as space-time shapes. *IEEE transactions on pattern analysis and machine intelligence*, **29**(12), 2247–53. 2.4.2.1, 4.3.3

Hardoon, D.R., Szedmak, S., & Shawe-Taylor, J. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, **16**(12), 2639–2664. 5.2.4

He, J., Kumar, S., & Chang, S.F. 2012. On the difficulty of nearest neighbor search. *Arxiv preprint arXiv:1206.6411*. 2.5.1.1

Helgason, S. 1999. *The radon transform*. Vol. 5. Birkhäuser Boston. 2.4.2.1

Holte, M.B., Moeslund, T.B., Nikolaidis, N., & Pitas, I. 2011a. 3d human action recognition for multi-view camera systems. *Pages 342–349 of: 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on*. IEEE. 2.4.2.2

Holte, Michael B, & Chakraborty, Bhaskar. 2011. A Local 3D Motion Descriptor for Multi-View Human Action Recognition from 4D Spatio-Temporal Interest Points. 1–13. 3.3.2

Holte, Michael B., Moeslund, Thomas B., Nikolaidis, Nikos, & Pitas, Ioannis. 2011b. 3D Human Action Recognition for Multi-view Camera Systems. *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, May, 342–349. 3.3.2

Hongeng, Somboon, & Nevatia, Ramakant. 2003. Large-scale event detection using semi-hidden Markov models. *Pages 1455–1462 vol.2 of: Proceedings Ninth IEEE International Conference on Computer Vision*, vol. 2. IEEE. 2.5.1.2

Hu, Weiming, Tan, Tieniu, Wang, Liang, & Maybank, Steve. 2004. A Survey on Visual Surveillance of Object Motion and Behaviors. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, **34**(3), 334–352. 2.2

Huang, J., & Zhang, T. 2010. The benefit of group sparsity. *The Annals of Statistics*, **38**(4), 1978–2004. 4.2

Hughes, G. 1968. On the mean accuracy of statistical pattern recognizers. *Information Theory, IEEE Transactions on*, **14**(1), 55–63. 2.4.4

Ikizler, Nazli, Cinbis, R. Gokberk, & Duygulu, Pinar. 2008. Human action recognition with line and flow histograms. *2008 19th International Conference on Pattern Recognition*, Dec., 1–4. 2.4.2.1

Iosifidis, Alexandros, Tefas, Anastasios, Nikolaidis, Nikolaos, & Pitas, Ioannis. 2012. Multi-view human movement recognition based on fuzzy distances and linear discriminant analysis. *Computer Vision and Image Understanding*, **116**(3), 347–360. 3.3.2

Jhuang, H, Serre, T, Wolf, L, & Poggio, T. 2007. A Biologically Inspired System for Action Recognition. *Pages 1–8 of: 2007 IEEE 11th International Conference on Computer Vision*. IEEE. 2.4.2.2

Kadir, T., & Brady, M. 2001. Saliency, scale and image description. *International Journal of Computer Vision*, **45**(2), 83–105. 2.4.3

Karthikeyan, S., Gaur, Utkarsh, Manjunath, B.S., & Grafton, Scott. 2011. Probabilistic subspace-based learning of shape dynamics modes for multi-view action recognition. *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, Nov., 1282–1286. 3.3.2

Kautz, H., & Allen, J.F. 1986. Generalized plan recognition. *Page 86 of: Proceedings of the fifth national conference on artificial intelligence*, vol. 19. Philadelphia, PA. 3.3.1

Ke, Yan, Sukthankar, Rahul, & Hebert, Martial. 2007. Event Detection in Crowded Videos. *Pages 1–8 of: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. 2.4.2.1

Kim, Tae-kyun, & Cipolla, Roberto. 2009. Canonical Correlation Analysis of Video Volume Tensors for Action Categorization and Detection. *Analysis*, **31**(8), 1415–1428. 2.4.2.1

Klaeser, A., Marszalek, Marcin, & Schmid, Cordelia. 2008. A Spatio-Temporal Descriptor Based on 3D-Gradients. *Pages 99.1–99.10 of: Procedings of the British Machine Vision Conference 2008*. British Machine Vision Association. 2.4.3

Laptev, Ivan. 2005. On Space-Time Interest Points. *International Journal of Computer Vision*, **64**(2-3), 107–123. 2.4.3

Laptev, Ivan, & Lindenberg, Tony. 2006. Local descriptors for spatio-temporal recognition. *Spatial Coherence for Visual Motion Analysis*. 2.4.3

Lee, H., Battle, A., Raina, R., & Ng, A.Y. 2007. Efficient sparse coding algorithms. *Advances in neural information processing systems*, **19**, 801. 7.1.3

Lewandowski, M, Makris, D, & Nebel, J C. 2010a. View and style-independent action manifolds for human activity recognition. *Computer Vision–ECCV 2010*, 547–560. 2.4.2.1

Lewandowski, Michal, Martinez-del Rincon, Jesús, Makris, Dimitrios, & Nebel, Jean-Christophe. 2010b. Temporal Extension of Laplacian Eigenmaps for Unsupervised Dimensionality Reduction of Time Series. *Pages 161–164 of: 20th International Conference on Pattern Recognition*. 2.4.4

Liao, L., Fox, D., & Kautz, H. 2007. Extracting Places and Activities from GPS Traces Using Hierarchical Conditional Random Fields. *The International Journal of Robotics Research*, **26**(1), 119–134. 2.5.1.2

Liggins, M.E., Hall, D.L., & Llinas, J. 2008. *Handbook of multisensor data fusion: theory and practice*. Vol. 22. CRC. 3, 3.1

Liu, Jingen, & Shah, Mubarak. 2008. Learning human actions via information maximization. *Pages 1–8 of: 2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2.4.4

Lu, W.L., Okuma, K., & Little, J.J. 2009. Tracking and recognizing actions of multiple hockey players using the boosted particle filter. *Image and Vision Computing*, **27**(1), 189–205. 2.4.2.1

Määttä, Tommi, & Aghajan, Hamid. 2010. On efficient use of multi-view data for activity recognition. 158–165. 3.3.2

Marr, D., & Nishihara, H.K. 1978. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, **200**(1140), 269–294. 2.4.1

Martinez-Contreras, F., Orrite-Urunuela, C., Herrero-Jaraba, E., Ragheb, H., & Velastin, S.A. 2009. Recognizing human actions using silhouette-based HMM. *Pages 43–48 of: Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*. Ieee. 2.4.4

Masoud, Osama, & Papanikolopoulos, Nikos. 2003. A method for human action recognition. *Image and Vision Computing*, **21**(8), 729–743. 2.4.2.2

Moeslund, Thomas B., & Granum, Erik. 2001. A Survey of Computer Vision-Based Human Motion Capture. *Computer Vision and Image Understanding*, **81**(3), 231–268. 2.2

Moeslund, Thomas B., Hilton, Adrian, & Krüger, Volker. 2006. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, **104**(2-3), 90–126. 2.2

Morency, L.P., Quattoni, A., & Darrell, T. 2007. Latent-dynamic discriminative models for continuous gesture recognition. *Pages 1–8 of: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. Ieee. 2.5.1.2, 7.1.2

Murphy, K.P. 2002. *Dynamic bayesian networks: representation, inference and learning*. Ph.D. thesis, University of California. 2.5.1.2

Nagel, H.H. 1988. From image sequences towards conceptual descriptions. *Image and vision computing*, **6**(2), 59–74. **??**, 2.3

Naiel, Mohamed A, & Abdelwahab, Moataz M. 2010. Multi-view Human Action Recognition System Employing 2DPCA Motaz El-Saban. 270–275. 3.3.2

Natarajan, Pradeep, Nevatia, Ramakant, Systems, Intelligent, & Angeles, Los. Coupled Hidden Semi Markov Models for Activity Recognition. 2.5.1.2

Ng, A.Y. 2004. Feature selection, L 1 vs. L 2 regularization, and rotational invariance. *Page 78 of: Proceedings of the twenty-first international conference on Machine learning*. ACM. 4.2

Nguyen, Nam T, Phung, Dinh Q, & Venkatesh, Svetha. 1987. Learning and Detecting Activities from Movement Trajectories Using the Hierarchical Hidden Markov Model. 2.5.1.2

Niebles, Juan Carlos, Wang, Hongcheng, & Fei-Fei, Li. 2008. Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. *International Journal of Computer Vision*, **79**(3), 299–318. 2.5.1.3

Niyogi, XH. 2004. Locality preserving projections. *Page 153 of: Advances in neural information processing systems 16: proceedings of the 2003 conference*, vol. 16. The MIT Press. 2.4.4, 5.1

Nowozin, Sebastian, Bakir, Gokhan, & Tsuda, Koji. 2007. Discriminative Subsequence Mining for Action Classification. *2007 IEEE 11th International Conference on Computer Vision*, 1–8. 2.5.1.3

Oikonomopoulos, Antonios, Patras, Ioannis, & Pantic, Maja. 2006. Spatiotemporal Salient Points for Visual Recognition of Human Actions. **36**(3), 710–719. 2.4.3

Oikonomopoulos, Antonios, Pantic, Maja, & Patras, Ioannis. 2009. Sparse B-spline polynomial descriptors for human activity recognition. *Image and Vision Computing*, **27**(12), 1814–1825. 2.4.3

Oliver, Nuria M, Rosario, Barbara, Pentland, Alex P, & Member, Senior. 2000. A Bayesian Computer Vision System for Modeling Human Interactions. **22**(8), 831–843. 2.5.1.2

Park, Sangho, & Aggarwal, J. K. 2004. A hierarchical Bayesian network for event recognition of human actions and interactions. *Multimedia Systems*, **10**(2), 164–179. 2.4.1

Pearson, K. 1901. Principal components analysis. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, **6**(2), 559. 2.4.4, 5.1, 5.1.4

Pehlivan, Selen, & Duygulu, Pınar. 2011. A new pose-based representation for recognizing actions from multiple cameras. *Computer Vision and Image Understanding*, **115**(2), 140–151. 3.3.2

Peng, Bo, Qian, Gang, & Rajko, Stjepan. 2009. View-invariant full-body gesture recognition via multilinear analysis of voxel data. *2009 Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, Aug., 1–8. 3.3.2, **??**, **??**, 6.5.3

Pérez, Ó., Piccardi, M., García, J., Patricio, M., & Molina, J. 2007. Comparison between genetic algorithms and the baum-welch algorithm in learning hMMs for human activity classification. *Applications of Evolutionary Computing*, 399–406. 2.5.1.2

Piccardi, M. 2004. Background subtraction techniques: a review. *Pages 3099–3104 of: Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 4. IEEE. 2.4.2.1

Piccardi, M., & Pérez, Ó. 2007. Hidden markov models with kernel density estimation of emission probabilities and their use in activity recognition. *Pages 1–8 of: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE. 2.5.1.2

Poppe, Ronald. 2010. A survey on vision-based human action recognition. *Image and Vision Computing*, **28**(6), 976–990. 2.2

Pruteanu-Malinici, Iulian, & Carin, Lawrence. 2008. Infinite hidden Markov models for unusual-event detection in video. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, **17**(5), 811–22. 2.5.1.2

Quattoni, Ariadna, Wang, Sybor, Morency, Louis-Philippe, Collins, Michael, & Darrell, Trevor. 2007. Hidden conditional random fields. *IEEE transactions on pattern analysis and machine intelligence*, **29**(10), 1848–53. 1.3, 2.5.1.2, 4.1

Rabiner, L.R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, **77**(2), 257–286. 2.5.1.2, 6.4, 6.4

Ragheb, Hossein, Velastin, Sergio, Remagnino, Paolo, & Ellis, Tim. 2008. Human action recognition using robust power spectrum features. *Pages 753–756 of: Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. 2.4.2.1

Ramagiri, S., Kavi, R., & Kulathumani, V. 2011. Real-time multi-view human action recognition using a wireless camera network. *2011 Fifth ACM/IEEE International Conference on Distributed Smart Cameras*, Aug., 1–6. 3.3.2

Rapantzikos, Konstantinos, Avrithis, Yannis, & Kollias, Stefanos. 2007. Spatiotemporal saliency for event detection and representation in the 3D Wavelet Domain: Potential in human action recognition. *Pages 294–301 of: Proceedings of the 6th ACM international conference on Image and video retrieval*. 2.4.2.2

Ribeiro, Pedro Canotilho, & Santos Victor, José. 2005. Human Activity Recognition from Video: modeling, feature selection and classification architecture. *In: Inter-*

national Workshop on Human Activity Recognition and Modelling HAREM 2005. 2.4.1, 2.4.2.2

Rudoy, Dmitry, & Zelnik-Manor, Lihi. 2011. Viewpoint Selection for Human Actions. *International Journal of Computer Vision*, **97**(3), 243–254. 3.3.2

Schindler, Konrad, & Gool, Luc Van. 2008. Action Snippets: How many frames does human action recognition require? *Pages 1–8 of: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. 2.4.2.2

Schmidt, M. 2010. *Graphical model structure learning with l1-regularization*. Ph.D. thesis, University of British Columbia. 4.2.1, 4.2.1

Schölkopf, B., Smola, A., & Müller, K.R. 1997. Kernel principal component analysis. *Artificial Neural Networks—ICANN'97*, 583–588. 2.4.4, 5.1.4

Schuldt, C., Laptev, Ivan, & Caputo, Barbara. 2004. Recognizing human actions: a local SVM approach. *Pages 32–36 Vol.3 of: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 00. IEEE. 2.5.1.3

Scovanner, Paul. 2007. A 3-Dimensional SIFT Descriptor and its Application to Action Recognition. *Comparative and General Pharmacology*, 1–4. 2.4.3

Shen, Changsong, Zhang, Chris, & Fels, Sidney. 2007. A Multi-Camera Surveillance System that Estimates Quality-of-View Measurement. *2007 IEEE International Conference on Image Processing*, III – 193–III – 196. 3.3.2

Smola, A.J., & Schölkopf, B. 1998. *Learning with kernels*. Citeseer. 5.1.3

Soro, Stanislava, & Heinzelman, Wendi. 2009. A Survey of Visual Sensor Networks. *Advances in Multimedia*, **2009**, 1–21. 1.1

Souvenir, R, & Babbs, J. 2008. Learning the viewpoint manifold for action recognition. *Pages 1–7 of: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2.4.2.1, 2.4.4

Srivastava, Gaurav, Iwaki, Hidekazu, Park, Johnny, & Kak, Avinash C. 2009. Distributed and lightweight multi-camera human activity classification. *2009 Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, Aug., 1–8. 3.3.2, **??**, 6.1, **??**

Steinberg, A.N., Bowman, C.L., & White, F.E. 1998. *Revisions to the JDL data fusion model*. American inst of aeronautics and astronautics new york. 3.1, 3.2.1

Sutton, Charles, Mccallum, Andrew, & Rohanimanesh, Khashayar. 2007. Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data. *Journal of Machine Learning Research*, **8**, 693–723. 2.5.1.2, 7.1.2

Szabó, Z., Póczos, B., & Lorincz, A. 2011. Online group-structured dictionary learning. *Pages 2865–2872 of: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 4.2

Talwalkar, A. 2010. *Matrix approximation for large-scale learning*. Ph.D. thesis, New York University. 5.4.1

Talwalkar, A., Kumar, S., & Rowley, H. 2008. Large-scale manifold learning. *Pages 1–8 of: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 5.3.2

Tavli, Bulent, Bicakci, Kemal, Zilan, Ruken, & Barcelo-Ordinas, Jose M. 2011. A survey of visual sensor network platforms. *Multimedia Tools and Applications*, **60**(3), 689–726. 1.1

Tenenbaum, J.B., De Silva, V., & Langford, J.C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, **290**(5500), 2319–2323. 2.4.4, 5.1, 5.1.5

Tran, D., & Sorokin, A. 2008. Human activity recognition with metric learning. *Computer Vision–ECCV 2008*, 548–561. (document), 2.4.2.2, 2.4.2.2, 2.4.4, 5.4.1, 6.2.1, **??**, C.1, C.1

Turaga, Pavan, Chellappa, Rama, Subrahmanian, V.S., & Udrea, Octavian. 2008. Machine Recognition of Human Activities: A Survey. *IEEE Transactions on Circuits and Systems for Video Technology*, **18**(11), 1473–1488. 2.2

Turaga, Pavan, Veeraraghavan, Ashok, & Chellappa, Rama. 2009. Unsupervised view and rate invariant clustering of video sequences. *Computer Vision and Image Understanding*, **113**(3), 353–371. 2.5.1.2

Vail, Douglas L., Veloso, Manuela M., & Lafferty, John D. 2007. Conditional random fields for activity recognition. *Page 1 of: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems - AAMAS '07*. New York, New York, USA: ACM Press. 2.5.1.2

van Kasteren, T.L.M., Englebienne, G., & Kröse, B.J.A. 2010. An activity monitoring system for elderly care using generative and discriminative models. *Personal and ubiquitous computing*, **14**(6), 489–498. 2.5.1.2

Vintsyuk, TK. 1968. Speech discrimination by dynamic programming. *Cybernetics and Systems Analysis*, **4**(1), 52–57. 2.5.1.1

Wang, Heng, Ullah, Muhammad Muneeb, Klaser, Alexander, Laptev, Ivan, & Schmid, Cordelia. 2009. Evaluation of local spatio-temporal features for action recognition. *Pages 124.1–124.11 of: Proceedings of the British Machine Vision Conference*. 2.4.3

Wang, Jack M, Fleet, David J, & Hertzmann, Aaron. 2008a. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, **30**(2), 283–98. 2.5.1.2

Wang, Liang, & Suter, David. 2007. Learning and matching of dynamic shape manifolds for human action recognition. *Image Processing, IEEE Transactions on*, **16**(6), 1646– 1661. 2.5.1.1

Wang, Liang, & Suter, David. 2008. Visual learning and recognition of sequential data manifolds with applications to human movement analysis. *Computer Vision and Image Understanding*, **110**(2), 153–172. 2.4.2.1, 2.4.4

Wang, Liang, Geng, Xin, Leckie, Christopher, Kotagiri, Ramamohanarao, Engineering, Software, & Technology, Information. 2008b. Moving Shape Dynamics: A Signal Processing Perspective. 0–7. 2.4.4

Wang, Yang, & Mori, Greg. 2008a. Learning a Discriminative Hidden Part Model for Human Action Recognition. *Pages 1721–1728 of: Advances in Neural Information Processing Systems (NIPS)*. 2.5.1.2

Wang, Yang, & Mori, Greg. 2008b. Max-Margin Hidden Conditional Random Fields for Human Action Recognition. 2.5.1.2

Wang, Ying, Huang, Kaiqi, & Tan, Tieniu. 2007a. Human Activity Recognition Based on R Transform. *Pages 1–8 of: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. 2.4.2.1

Wang, Ying, Huang, Kaiqi, & Tan, Tieniu. 2007b. Multi-view Gymnastic Activity Recognition with Fused HMM. 667–677. 3.3.2

Weinland, Daniel, Ronfard, Remi, & Boyer, Edmond. 2006. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, **104**(2-3), 249–257. 2.4.2.1, **??**, **??**, B.2

Weinland, Daniel, Ronfard, Remi, & Boyer, Edmond. 2011. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, **115**(2), 224–241. 2.2

White, F., *et al.* . 1988. A model for data fusion. *Pages 149–158 of: Proc. 1st National Symposium on Sensor Fusion*, vol. 2. 3.1, 3.2.1

Willems, Geert, Tuytelaars, Tinne, & Gool, Luc Van. 2008. An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector. *Pages 650–663 of: Computer Vision–ECCV 2008*. 2.4.3

Williams, C.K.I., & Seeger, M. 2001. Using the Nystrom method to speed up kernel machines. *Advances in neural information processing systems*, 682–688. 5.3.2

Wong, Shu-Fai, & Cipolla, Roberto. 2007. Extracting Spatiotemporal Interest Points using Global Information. *Pages 1–8 of: 2007 IEEE 11th International Conference on Computer Vision*. IEEE. 2.4.3

Wong, Shu-Fai, Kim, Tae-Kyun, & Cipolla, Roberto. 2007. Learning Motion Categories using both Semantic and Structural Information. *Pages 1–6 of: 2007 IEEE Conference on Computer Vision and Pattern Recognition*. Ieee. 2.5.1.3

Wu, Chen, Khalili, Amir Hossein, & Aghajan, Hamid. 2010. Multiview activity recognition in smart homes with spatio-temporal features. *Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras - ICDSC '10*, 142. 3.3.2

Wu, Tsu-yu, Lian, Chia-chun, & Hsu, Jane Yung-jen. 2001. Joint Recognition of Multiple Concurrent Activities using Factorial Conditional Random Fields. 2.5.1.2

Xiang, Tao, & Gong, Shaogang. 2006. Beyond Tracking: Modelling Activity and Understanding Behaviour. *International Journal of Computer Vision*, **67**(1), 21–51. 2.4.2.1, 2.5.1.2

Xiang, Tao, & Gong, Shaogang. 2008. Optimising dynamic graphical models for video content analysis. *Computer Vision and Image Understanding*, **112**(3), 310–323. 2.5.1.2

Yamato, Junji, Ohya, Jun, & Laboratories, Human Interface. 1992. Recognizing Human Action in Time-Sequential Images using Hidden Markov Model. *Pages 379–385 of: Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*. 2.4.2.1

Yan, Pingkun, Khan, Saad M., & Shah, Mubarak. 2008. Learning 4D action feature models for arbitrary view action recognition. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, June, 1–7. 3.3.2

Yan, S., Xu, D., Zhang, B., Zhang, H.J., Yang, Q., & Lin, S. 2007. Graph embedding and extensions: A general framework for dimensionality reduction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **29**(1), 40–51. 5.1, 6.2.2

Yang, Yuedong, Hao, Aimin, & Zhao, Qinping. 2008. View-invariant action recognition using interest points. *Proceeding of the 1st ACM international conference on Multimedia information retrieval - MIR '08*, 305. 2.5.1.3

Yilmaz, A, & Shah, M. 2005. Actions sketch: A novel action representation. *Pages 984–989 of: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE. 2.4.2.1

Yilmaz, Alper, Javed, Omar, & Shah, Mubarak. 2006. Object tracking. *ACM Computing Surveys*, **38**(4), 13–es. 2.1, 2.4.1

Zhang, Jianguo, & Gong, Shaogang. 2010a. Action categorization by structural probabilistic latent semantic analysis. *Computer Vision and Image Understanding*, **114**(8), 857–864. 2.5.1.3

Zhang, Jianguo, & Gong, Shaogang. 2010b. Action categorization with modified hidden conditional random field. *Pattern Recognition*, **43**(1), 197–203. 2.5.1.2

Zhang, Ziming, Hu, Yiqun, Chan, Syin, & Chia, Liang-tien. 2008. Motion Context: A New Representation for Human Action Recognition. 817–829. 2.4.2.2

Zhu, C., Byrd, R.H., Lu, P., & Nocedal, J. 1997. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, **23**(4), 550–560. 4.1.1

Zhu, Fan, Shao, Ling, & Lin, Mingxiu. 2012. Multi-View Action Recognition Using Local Similarity Random Forests and Sensor Fusion. *Pattern Recognition Letters*, May. 3.3.2

Zhu, Yan, Zhao, Xu, & Fu, Yun. 2011. Sparse coding on local spatial-temporal volumes for human action recognition. *Pages 660–671 of: Computer Vision - ACCV 2010*. 2.4.4