Model free approach towards human action recognition

A dissertation submitted by **Bhaskar Chakraborty** at Universitat Autònoma de Barcelona to fulfil the degree of **Doctor en Informàtica**.

Bellaterra, July 2012

Director	**Dr. Jordi Gonzàlez i Sabaté** Centre de Visió per Computador Dept. de Ciències de la Computació, Universitat Autònoma de Barcelona.
Co-director	**Dr. Xavier Roca i Marvà** Dept. de Ciències de la Computació, Universitat Autònoma de Barcelona. Centre de Visió per Computador
Thesis Committee	**Dr. Cristian Sminchisescu** Institue für Numerische Simulation Universität Bonn **Dr. Sergio Escalera** Dept. Matemática Aplicada i Análisi Universitat de Barcelona **Dr. Nicolas Pérez de la Blanca** Ciencias de la Computación e I.A. ETSI Informática y de Telecomunicación Universidad de Granada **Dr. Manuel Jesús Marín Jiménez** Informática y Análisis Numéricos Universidad de Córdoba **Dr. Caifeng Shan** School of Electronic Engineering and Computer Science Queen Mary University of London

This document was typeset by the author using LaTeX 2ε.

The research described in this book was carried out at the Computer Vision Center, Universitat Autònoma de Barcelona.

Copyright © 2012 by Bhaskar Chakraborty. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the author.

ISBN 978-84-940231-6-8

Printed by Ediciones Gráficas Rey, S.L.

To my parents.

Acknowledgements

The research work presented in this thesis is an outcome of a long duration process of hard work, through which I have this opportunity to meet and collaborate a number of kind and extraordinary people, without whom it would not have been possible to continue my journey as a researcher.

First of all, I would like to thank to my thesis supervisors, Dr. Jordi Gonzàlez, Dr. Xavier Roca and Dr. Juan José Villanueva. To begin with, thanks to Dr. Juan José to give me this opportunity to come to Barcelona and fulfill my dream to complete the PhD. Thanks to Xavi for his great support and advice. Finally, Jordi would always be a special person to be remembered. He is somewhat my friend-philosopher-guide. Without his guidance, patience and faith this thesis would not be completed.

I would also like to thank to Dr. Ignasi Rius and Dr. Andrew D. Bagdanov as my second supervisors during my PhD. Ignasi's help during the initial phase of my PhD has been invaluable. Andy's comments and opinions have always showed me a path to represent better my work. I would like to express my gratitude to Dr. Thomas B. Moeslund, Dr. Micheal B. Holte and Dr. Preben Fhil for helping me during my research stay in Aalborg University, Denmark.

My special thanks to the thesis jury members: Dr. Cristian Sminchisescu, Dr. Nicolas Pérez de la Blanca and Dr. Sergio Escalera for their presence in Computer Vision Center as my thesis evaluators. I would also like to thank to all the administrative stuffs, specially Montse, Gigi and Mari, for helping me out with any bureaucracy work and allow me to concentrate on my research. I also acknowledge the support from Agéncia de Gestio d'Ajuts Universiteris i Recerca (AGAUR) Generalitat de Catalunya, for granting me the predoctoral and mobility scholarships.

Allow me this opportunity to thank my dearest collegues at CVC. Marco, Ariel and Murad are most remembered ones as we started this research adventure together. My thanks to Ivan, Pau and Carles to make me feel at home and extending their priceless friendship. My thanks to German, Pep, Javier Vasquez, Xu Hu and Onur for their nice company and discussion. I would like to thank to Dr. Francesco Ciompi and Dr. Pierluigi Casale as well, for their friendship and support. My special thanks to Dr. Partha Pratim Roy and Anjan Dutta for their company, which always bring me back to the nice memories of my homeland.

Apart from the research environment, I would like to thank to my local friends: Pablo, Dani, Sandra, Javi and Pablito, for making me feel Barcelona as my second home.

Finally my thanks, love and respect to those persons, very close to my heart, my family in India. My thanks and gratitude to my parents and sister, for their unequivocal support throughout, as always, for which my mere expression of thanks would never be enough. My special thanks to Marina, for her love, care and support every day.

Abstract

Automatic understanding of human activity and action is a very important and challenging research area of Computer Vision with wide scale applications in video surveillance, motion analysis, virtual reality interfaces, robot navigation and recognition, video indexing, content based video retrieval, HCI, health care, choreography and sports video analysis etc. This thesis presents a series of techniques to solve the problem of human action recognition in video. First approach towards this goal is based on the a probabilistic optimization model of body parts using hidden markov model (HMM). This strong model based approach is able to distinguish between similar actions by only considering the body parts having major contributions to the actions, for example legs for walking and jogging; arms for boxing and clapping. Next approach is based on the observation that the action recognition can be done using only the visual cue, i.e. human pose during the action, even with the information of few frames instead of examining the whole sequence. In this method, actions are represented by a Bag-of-*key-poses* model to capture the human pose variation during an action.

To tackle the problem of recognizing the action in complex scenes, we propose a model free approach which is based on the Spatio-temporal interest point (STIP) and local feature. To this end, a novel STIP detector is proposed which uses a mechanism similar to that of the non-classical receptive field inhibition that is exhibited by most orientation selective neurons in the primary visual cortex. An extension of the selective STIP based action recognition is applied to the human action recognition in multi-camera system. In this case, selective STIPs from each camera view point are combined using the 3D reconstructed data, to form 4D STIPs [$3D$ space + time] for multi-view action recognition. The concluding part of the thesis dedicates to the continuous visual event recognition (CVER) on large scale video dataset. This is an extremely challenging problem due to high scalability, diverse real environment state and wide scene variability. To address these issues, a motion region extraction technique is applied as a preprocessing step. A max-margin generalized Hough Transform framework is used to learn the feature vote distribution around the activity center to obtain an activity hypothesis which is verified by a Bag-of-words + SVM action recognition system. We validate our proposed approaches on several benchmark action recognition datasets as well as small scale and large scale activity recognition datasets. We obtain state-of-the results which shows a progressive improvement of our proposed techniques to solve human action and activity recognition in video.

Resumen

La comprensión automática de las acciones humanas observadas en secuencias de imágenes es muy importante en el área de investigación de la Visión por Computador, con aplicaciones a gran escala en la vigilancia de vídeo, análisis del movimiento humano, interfaces de realidad virtual, robots de navegación, así como para el reconocimiento, indexación, y recuperación de vídeo. Esta tesis presenta una serie de técnicas para resolver el problema del reconocimiento de las acciones humanas en video. Nuestro primer enfoque hacia esta tema se basa en la optimización de un modelo probabilstico de las partes del cuerpo utilizando una Hidden Markov Model (HMM). Este enfoque se basa en un *strong model*, capaz de distinguir entre acciones similares considerando sólo las partes del cuerpo que tienen las mayores aportaciones en la realización de ciertas acciones, por ejemplo en piernas para *caminar* y *correr*, o en brazos para acciones como *boxeo* y *aplaudir*. Nuestro siguiente enfoque se basa en la observación de que el reconocimiento de acciones se puede realizar usando sólo información visual, i.e. *la postura humana* desarrollada durante una acción, analizando la la informacin de unos cuantos frames en lugar de examinar la secuencia completa. En este método, las acciones se representan mediante un modelo Bag-of-*key-poses* para poder capturar la variación de la postura humana durante el desarrollo de una acción.

Para hacer frente al problema del reconocimiento de la acción en escenas complejas, a continuación se propone una aproximación *model free* basada en el análisis de puntos de interés espacio-temporales (STIPs) que disponen de mucha información local. Para este fin, se ha desarrollado un nuevo detector de STIPs que se basa en el mecanismo de inhibición del campo receptivo utilizado en la corteza primaria, en particular en la orientación selectiva visual de las neuronas. Además, hemos extendido nuestro reconocimiento de acciones basado en STIPs selectivos a sistemas multi-cámara. En este caso, los STIPs selectivos de cada punto de vista se combinan mediante los datos 3D reconstruidos para formar STIPs selectivos 4D (espacio 3D + tiempo).

En la parte final de esta tesis, nos dedicamos al reconocimiento continuo de eventos visuales (CVER) en bases de datos de vídeos de seguridad enormes, con un gran conjunto de datos. Este problema es extremadamente difícil debido a la alta escalabilidad de los datos, a las dificultades del entorno real en el que se aplcia y a una variabilidad en escena muy amplio. Para abordar estos problemas, las regiones en movimiento son detectadas a partir de una técnica llamada *max margin generalized Hough transformation*, que se utiliza para aprender aquella distribución de características entorno a una acción para reconocer hipótesis que luego se verifican por Bag-of-words más un clasificador lineal. Hemos validado nuestras técnicas en varios conjuntos de datos de video vigilancia que constituyen el estado del arte actual en este tema. Los resultados obtenidos demuestran que hemos mejorado la precisión en la detección de acciones humanas en vídeo.

Resum

La comprensió automática de les accions humanes observades en seqüéncies d'imatges és molt important en el área de recerca de la Vision per Computador, amb aplicacions a gran escala en la vigiláncia de vídeo, análisi del moviment humá, interfícies de realitat virtual, robots de navegació, així com per al reconeixement, indexació, i recuperació de vídeo. Aquesta tesi presenta una sèrie de tècniques per resoldre el problema del reconeixement de les accions humanes en vídeo. El nostre primer enfocament cap a aquesta tema es basa en la optimització d'un model probabilístic de les parts del cos utilitzant una Hidden Markov Model (HMM). Aquest enfocament es basa en un *strong model*, capaç de distingir entre accions similars considerant només les parts del cos que tenen les majors aportacions en la realització de certes accions, per exemple en cames per caminar i córrer, o en braços per a accions com boxa i aplaudir. El nostre següent enfocament es basa en l'observació de que el reconeixement d'accions es pot realitzar usant només informació visual, ii la postura humana desenvolupada durant una acció, analitzant la la informació d'uns quants frames en lloc d'examinar la seqüéncia completa. En aquest métode, les accions es representen mitjanant un model Bag-of- textit key-poses per poder capturar la variació de la postura humana durant el desenvolupament d'una acció.

Per fer front al problema del reconeixement de l'acció en escenes complexes, tot seguit es proposa una aproximaci'o *model free* basada en l'anàlisi de punts d'interès espai-temporals (STIPs) que disposen de molta informació local. Amb aquesta finalitat, s'ha desenvolupat un nou detector de STIPs que es basa en el mecanisme de inhibició del camp receptiu utilitzat en l'escora primària, en particular en l'orientació selectiva visual de les neurones. A més, hem estès el nostre reconeixement d'accions basat en STIPs selectius a sistemes multi-càmera. En aquest cas, els STIPs selectius de cada punt de vista es combinen mitjanant les dades 3D reconstruts per formar STIPs selectius 4D (espai 3D + temps).

A la part final d'aquesta tesi, ens dediquem al reconeixement continu d'esdeveniments visuals (CVER) en bases de dades de vídeos de seguretat enormes, amb un gran conjunt de dades. Aquest problema és extremadament difcil a causa de l'alta escalabilitat de les dades, a les dificultats de l'entorn real en què es aplcia ja una variabilitat en escena molt ampli. Per abordar aquests problemes, les regions en moviment són detectades a partir d'una tècnica anomenada *max margin generalized Hough transformation*, que s'utilitza per aprendre aquella distribució de caracterstiques voltant d'una acció per reconèixer hipòtesis que després es verifiquen per Bag-of-words mes un classificador lineal. Hem validat les nostres tècniques en diversos conjunts de dades de vdeo vigilncia que constitueixen l'estat de l'art actual en aquest tema. Els resultats obtinguts demostren que hem millorat la precisió en la detecció d'accions humanes en vídeo.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

"Human beings must have action; and they will make it if they cannot find it."

Albert Einstein

Recognizing human activities from video is one of the most promising areas of computer vision. In recent years, this problem has caught the attention of researchers from many different communities: industry, academia, security agencies and consumer agencies. One of the earliest investigations into the nature of human motion was conducted by the contemporary photographers Etienne Jules Marey and Eadweard Muybridge in the 1850s who photographed moving subjects and revealed several interesting and artistic aspects involved in human and animal locomotion. The classic Moving Light Display (MLD) experiment of Johansson [83] provided a great impetus to the study and analysis of human motion perception in the field of neuroscience. This then paved the way for mathematical modeling of human action and automatic recognition, which naturally fall into the purview of computer vision and pattern recognition.

To state the problem in simple terms, given a video sequence with one or more persons performing an activity, can a system be designed that can automatically recognize what activity is being or was performed ? The question is quite straightforward, yet the solution is much harder to find. In a simple case where a video contains only one execution of a human activity, the objective of the system is to correctly classify the video into its activity category. In more general cases, the continuous recognition of human activities must be performed, detecting the start and end frames of all occurring activities from an input video.

There are various types of human activities. Depending on their complexity, these can be conceptually categorized into *four* different levels: gestures, actions, interactions, and group activities [2]. Gestures are elementary movements of a person's body part, and are the atomic components describing the meaningful motion of a person.

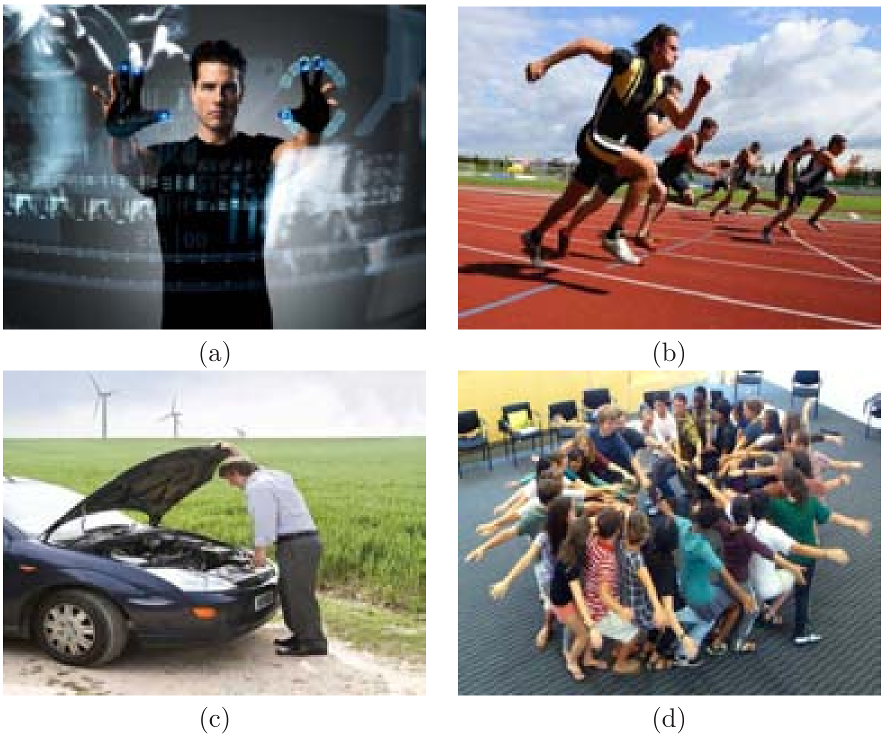

(a) (b) (c) (d)

Figure 1.1: Different types of human activities, (a) gesture, (b) action, (c) interaction and (d) group action. In this thesis, automatic recognition of *action* and *interaction* is addressed.

'Stretching an arm', 'raising a leg' and 'indicating directions' are simple examples of gestures. Actions are single person activities that may be composed of multiple gestures organized temporally, such as 'walking', 'waving', and 'hand shaking'. Some complex *actions* involve more than two person, like 'hand shaking'. Interactions are human activities that involve two or more persons and/or objects. For example, 'two persons fighting' is an interaction between two humans and 'a person getting inside a car' is a human-object interaction involving one human and one object. Finally, group activities are the activities performed by conceptual groups composed of multiple persons and/or objects. 'A group of persons marching', 'a group having a meeting', are typical examples of them.

This thesis work presents different approaches for recognizing *Actions* and *Interactions*. We focus on the problem of human action recognition in both single camera (2D) and and multi-camera (3D) environments. Different scene complexities starting from simple background to complex cluttered background, recording settings from static camera to moving camera and scene resolutions have been addressed. For *interaction*, we only focus on the human-car interactions, like 'person getting into the car', 'person opening the back trunk' etc.

1.1 Applications of Human action recognition

The ability to recognize complex human activities from videos enables the construction of several important applications [181].

1.1.1 Video surveillance

Security and surveillance systems have traditionally relied on network of video cameras monitored by a human operator who needs to be aware of the activity in the camera's field of view. With recent growth in the number of cameras and deployments automatic video surveillance is becoming more important. Automated surveillance systems in public places like airports and subway stations require detection of abnormal and suspicious activities as opposed to normal activities. For instance, an airport surveillance system must be able to automatically recognize suspicious activities like 'a person leaving a bag' or 'a person placing his/her bag in a trash bin'. Automatic recognition of human activities can also be applied in the real-time monitoring of patients, children, and elderly persons. The construction of gesture-based human computer interfaces and vision-based intelligent environments becomes possible as well with an activity recognition system. A related video surveillance application involves searching for an activity of interest in a large database by learning patterns of activity from long videos.

1.1.2 Content based video analysis

Video has become a part of our everyday life. The need for developing efficient indexing and storage schemes is increasing to improve user experience with the implacable growth in the video sharing websites. This requires learning of patterns from raw video and summarizing a video based on its content. A query processing system for a video library can have at its core, a human action recognition system which scans through video taking as input, an action-query specified in high-level language and producing as output, the sequence of video frames having that action-query. Such an application could prove very useful for sportscasters to quickly retrieve important events in particular games. For example, a query of finding 'direct free-kicks' in a football game or a 'cover drive shot' in a cricket match footage could be automatically retrieved with the specific frames where these actions occur. Such a system will eliminate the need for cumbersome manual annotation of video.

1.1.3 Human computer interaction

Understanding the interaction between human and computer remains an open research problem in designing human-computer interfaces. Visual cues are the most important mode of non-verbal communication. Effective utilization of this mode such as gestures and activities holds the promise of helping in creating computers that can better interact with humans. Similarly, interactive environment such as smart rooms that can react to a user's gestures can benefit from the automatic human action recognition methods.

1.1.4 Animation and Synthesis

The gaming and animation industry rely on synthesizing realistic humans and human motion. Human gesture recognition is becoming a fundamental part in more realistic gaming environment such as Microsoft Xbox using KINECT. With improvements in algorithms and hardware, much more realistic motion-synthesis is now possible. A related application is learning in simulated environments. Examples of this includes training of military soldiers, fire-fighters and other rescue personal in hazardous situations with simulated objects.

1.1.5 Behavioural biometrics

Biometrics involves study of the approached and algorithms for uniquely recognizing humans based on physical or behavioural cues. Traditional approaches are based on fingerprint, face or iris and can be classified as Physiological Biometric i.e. they rely or physical attributes for recognition. Recently, 'Behavioural Biometric' have been gaining popularity, where the main idea is to use behaviour as an useful cue to recognize humans as their physical attributes. Since observing behaviour implies longer-term observation of the subject, approaches for action-recognition extend naturally to this task.

1.1.6 Health care

Health sector is a very important area and application of vision based techniques are observed more and more in recent days. Activity recognition in indoor scenario can be used to design 'smart elderly care' system. This system can automatically analyze the activities of the elderly and react to the abnormal behaviour. Analysis of the movements of different body parts can also be applied to functional rehabilitation. For example, a patient performing certain physical exercise can obtain feedbacks through a movement recognition system.

1.2 Scope of the thesis

Human action recognition from video is a vast area and it is important to mention the types of action/activity recognition problems are addressed, along with the methodologies that are applied. The basic process of the task of human action recognition can be divided into three main issues:

- Finding region of interest where the action is happening.
- Representing the action.
- Classifying the action into its category.

In this thesis work, all these issues are elaborated. In particular, we have shown *three* different approaches for the first issue, i.e. ‘identification of the action regions’. The analysis begin with strong model based approach where human body parts are first identified and then their stochastic behaviour is modeled. Second framework uses filtering approach i.e. a weaker model to detect the body limb regions as ‘action areas’ and extracts *key-poses* using limb dynamics for action representation. This approach mainly explores the idea of utilizing pose variations to recognize actions. The last approach is based on the model free technique such as spatio-temporal interest points (STIPs) for as ‘action regions’. In this framework, a selective STIP detector is proposed using a surrounding suppression mask which keeps more repeatable and stable STIPs on the moving parts while suppressing the unwanted background interest points. Furthermore, STIP detector is applied to recognize human action recognition in multi-camera environment, where it is extended as $4D$ STIPs by taking $[X, Y, Z, t]$ axes into consideration and for large scale activity recognition.

The second issue i.e. action representation is mostly done using state-of-the-art methods. Action representation is actually refers to different features extracted from the action regions. For strong and weak model based approach we use histogram of oriented gradient [40] features. The goal is to investigate how is the overall shape of the moving region capable of distinguishing the action. To this end, directional statistics [192] are computed from the HOG feature to obtain a better representation of the pose variation. In the model free approach, popular N-jet features [159] are extracted from a cuboid neighbourhood of STIPs for action representation. For multi-camera human action recognition, 3D optical flow features [71] at the hyper-cuboid neighbourhood of $4D$ STIPs are used and lastly, a concatenation of histogram of 3D optical flow (HOF3D) [32], histogram of oriented gradient in 3D (HOG3D) [23] and extended SURF (ESURF) [202] is used to represent the activities.

In the action recognition part, several machine learning approaches are applied. In the strong model based approach Hidden Markov Model (HMM) is used. In the weak model based approach a Bag-of-key poses model is used along with the Support Vector Machine (SVM) classifier. For model free approach and human action recognition in multi-camera environment, a Bag-of-words model + Support Vector Machine (SVM) framework is applied. Finally, a max-margin generalized Hough transform technique is used for the large scale activity detection.

Our proposed algorithms as applied on the following benchmark datasets: HumanEva, HERMES (indoor), KTH, Weizmann, CVC action, YouTube, Holloywood2, Multi-KTH, CMU, MSR and VIRAT. These datasets are having different challenges: simple, semi-complex and complex background; static and moving camera, widely varying scene reconstitutions and human actors. Please refer to the Chapter A.

1.3 Contributions

In this thesis, we propose three main approaches for human action and activity recognition: strong model based approach, relatively weak model based approach and model free approach. In addition, methods for multi-camera human action recognition and activity detection in large scale are also introduced. Following are the contributions in each proposed research line.

- **Strong model based approach**:
    - A novel approach for recognizing actions based on a probabilistic optimization model of body limbs is proposed.
    - Viewpoint invariant human detection using different example-based body-part detectors are applied.
    - A novel feature selection technique is proposed for designing body-part detectors.
    - Stochastic movements of the body limbs having major contributions to the actions like legs for walking, jogging and running; arms for boxing, clapping and waving, are modeled using an Hidden Markov Model.
    - Our approach is trained using a baseline dataset HumanEva and applied to different benchmark datasets, KTH and Weizmann. This shows that our method is robust and generalizes to various datasets having different actions and illuminations.
- **Weak model based approach**
    - In this method we show that the limb dynamics feature based *key-poses* are sufficient for fast action recognition.
    - Limb regions are identified using a real time human detector based on an Average Synthetic Exact Filter. We extend the ASEF human detector in multi-scale frame work and introduce a verification SVM to obtain a robust real time human detection. We introduce a Directional HOG features are used to capture the limb dynamics i.e. the pose variations during the action.
    - *K-medoids* clustering technique is applied on the limb dynamics features to obtain the action *key-poses*. Action videos are represented using a Bag-of-*key-poses* model.
    - This method is fast, uses simple features and obtains state-of-the-art recognition rate on the benchmark action recognition dataset: KTH, Weizmann and CVC action.
- **Model free approach**
    - We introduce a novel approach for selective STIP detection, by applying surround suppression combined with local and temporal constrains, achieving robustness to camera motion and background clutter. The suppression

mask uses a mechanism similar to that of the non-classical receptive field inhibition that is exhibited by most orientation selective neurons in the primary visual cortex.

  - A novel Bag of Video words model of local N-jet features, computed at the detected STIPs, is proposed. This end a pyramid structure is applied to capture the spatial information and vocabulary compression is introduces at each pyramid level to reduce the dimensionality of the feature space.
  - An automatic video annotation technique is proposed using an actor specific spatio-temporal clustering of STIPs.
  - We evaluate our approach on both popular benchmark datasets (KTH and Weizmann), more challenging datasets (CVC, CMU), movie and YouTube video clips (Hollywood2 and YouTube) and perform an exhaustive cross-data evaluation, trained on source dataset (KTH and Weizmann) and tested on more challenging target datasets (CVC, CMU, MSR I and Multi-KTH).

- **Recognizing human action in multi-camera framework**

  - In this approach we extend the idea of the selective STIPs in multi-camera environment. This is done by combining selective STIPs from each camera view-point by using the 3D reconstruction model to obtain a novel $4D$ ($3D$ space + time) STIPs for action recognition.
  - A novel local view-invariant $3D$ motion descriptor, histogram of optical $3D$ flow, computed on the hyper-cuboid neighbourhood of each $4D$ STIP is proposed.
  - A pyramid BoV model with vocabulary compression is introduced to model different action.
  - Experiments on publicly available datasets, i3DPost and IXMAS are performed showing state-of-the-art performances.

- **Continuous visual event detection in large scale**

  - A generalized max-margin Hough transformation framework is introduced for the activity detection in large scale. This motivated by [123, 106] which are applied to $2D$ object recognition.
  - To reduce the initial search space, which is a major computational complexity issue for large scale activity detection, a region clustering based motion segmentation algorithm is proposed.
  - To make the Hough transformation based activity detection framework more robust, a verification action SVM is introduced.
  - The algorithm is tested both on large scale (VIRAT) and small scale (MSR) activity detection datasets.

1.4 Thesis structure

The thesis structure is as follows:

- Chapter 2 presents a brief review of the state-of-the art methods for human action recognition. This chapter gives an overview of different techniques of human action recognition starting from strong model based approaches to model free approaches.

- Chapter 3 describes an approach to human action recognition based on a probabilistic optimization of body parts using hidden Markov model (HMM). Here we present the strong model based approach to identify the action region.

- Chapter 4 introduces the idea of representing human action by using the *key-poses* obtained from the body limb dynamics. ASEF human detector is presented for a fast human identification. This chapter focuses on the weak model based human action recognition technique.

- Chapter 5 includes the description of a selective spatio-temporal interest point detector. In particular, we present a STIP detector for human action recognition in complex scenes, a model free approach to identify action region.

- Chapter 6 documents an application of the selective STIP into action recognition in multi-camera environment. Here we extend the previous STIP detector into a $4D$ counterpart by taking $[XYZt]$ axes into consideration.

- Chapter 7 presents the a max-margin Hough transformation framework for large scale continuous event detection. This shows another application of the selective STIP detector.

- Chapter 8 documents the conclusion and future work of the thesis.

Chapter 2

Related work

> *"Citations are an acknowledgement of intellectual debt. Web of Science lets researchers instantly recognize works that are well regarded by their peers. That way, they know they are basing their work on quality research."*
>
> *Fifty years of citation and indexing and analysis* (2005), by Henry Small

Visual event recognition has been an important research area of Computer Vision and a large variety of approaches are documented in the literature. In this chapter a thorough review of different state-of-the-art techniques is done by dividing the action recognition methods into four main directions: global representation based approaches, model free or local representation based approaches, methods for action recognition in multi-camera environment and activity detection approaches both in small and large scale.

The area of human action recognition is a continuously growing field of Computer Vision and an impressive collection of research works have been found in the literature. Several existing surveys document various action recognition approaches [1, 2, 55, 81, 129, 148, 181, 201]. Action recognition is a vast area of research and often several taxonomies like, 'action', 'activities', 'gestures' and 'motion' etc are categorized into action recognition. We will focus mainly the review of the methods dedicated to 'actions' and 'activities' (See Chapter 1). As discussed in the previous chapter (Chapter 1), action recognition techniques are addressed to solve three main issues, the detection of action region, action representation and finally the action recognition. The state-of-the-art approaches are usually focus to either of these issues or the combination of those.

Action recognition is not only restricted to the single camera (2D) environment, approaches to solve action recognition problems in multi-camera setup (3D) has also gained an immense interest. The methods for this area (3D) utilize the techniques for the $2D$ counterpart with some modification to fit with the $3D$ environment. Other area of action recognition is the activity detection, where apart from the action recognition the exact location of the action/activity must be identified.

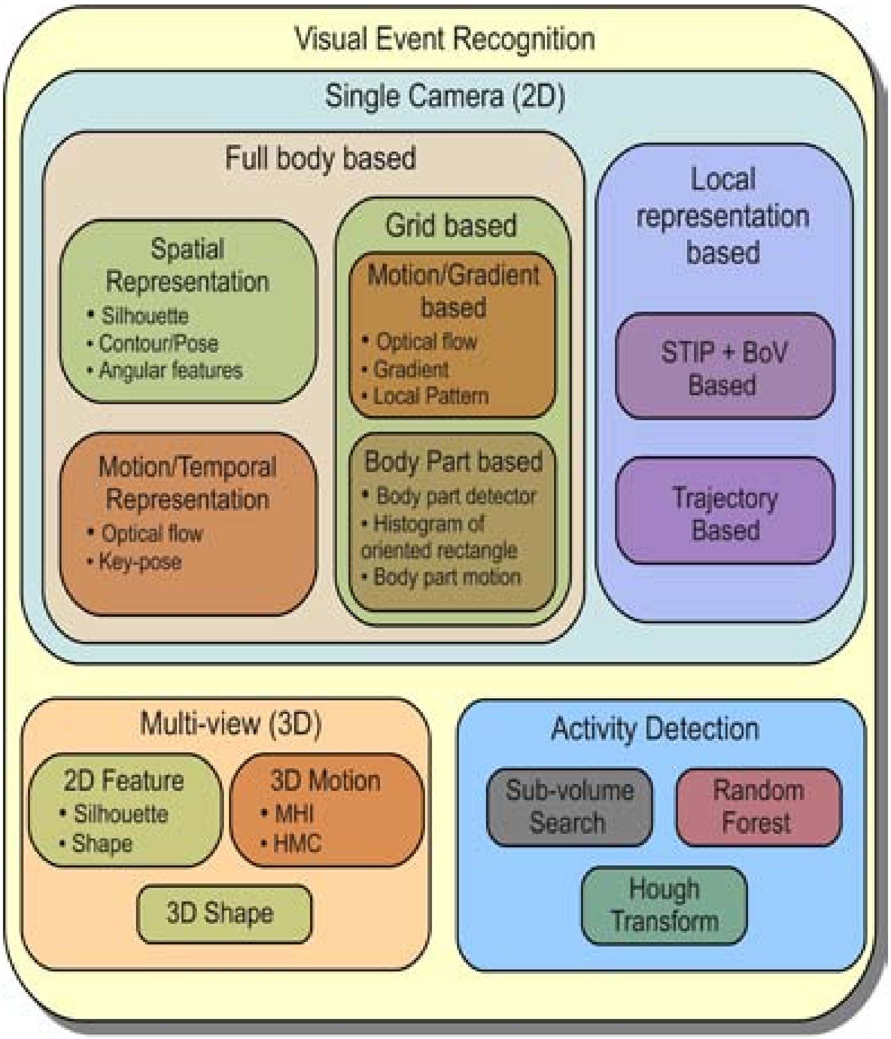


Figure 2.1: Action recognition taxonomy. As described in the text, visual event recognition is divided into three main categories: single camera (2D) and multi-view (3D) and activity detection. Each node of the tree represent different approaches that address the corresponding methods.

According to our thesis scope and structure we divide the state-of-the-art methodologies into *four* main groups. Section 2.1 describes the techniques based on full body or holistic features and these methods eventually apply strong or week models to detect and represent the action region of interest. In Section 2.2 state-of-the-art techniques based on the local representation of the action is described. Action recognition in multi-camera environment is presented in 2.3. Finally, methods for activity recognitions are discussed in Section 2.4.

2.1 Holistic body model based approaches

Human action recognition based on the holistic body model based approach relies on the global representation to encode the visual observation as a whole in a top down fashion. In this type of approaches, first the human is identified in the image frame by using background subtraction, tracking or person detector. Then the region of interest is encoded as a whole, which results in the image descriptor [148]. This is an intuitive and biologically-plausible approach to action recognition, which is supported by psychophysical work on visual interpretation of biological motion [83]. The approaches in this category use the spatial representation (silhouette, contours, angles and key-poses) and motion representation (optical flow) of the actions. Grids are also applied to divide the global features into parts to obtain more robustness to noise and view-point changes.

2.1.1 Spatial representation:

In this category, the main focus is to extract image features that are discriminative with respect to pose of the human body, as visual cue, distinguish actions.

Silhouette based

One of the earliest uses of silhouette for action recognition is by Bobick et al. [16]. In this approach, silhouettes are extracted from a single view and the aggregated differences between subsequent frames of an action sequence. This results in a binary motion energy image (MEI) which indicates the motion occurrence. Also, a motion history image (MHI) is constructed where pixel intensities are a recency function of the silhouette motion. Other approach for the silhouette extraction [193] use $\Re$ transform which results in a translation and scale invariant representation. Souvenir et al. [172] extend the $\Re$ transformation computation in time domain. Weinland et al. [198] match two silhouettes using Euclidean distance for action recognition. Action recognition by recovering 2D stick figure from the skeleton of the human silhouette is proposed by Guo et al. [66]. Niyogi et al. detect a stick figure from the space time volume spanned by an image sequence of a walking person [136].

Yamato et al. [206] first introduce the use of hidden Markov model (HMM), where silhouette images are quantized into super-pixels and each pixel counts the ratio of black and white pixels within its underlying region as features. Ahmad et al. [4] use human body silhouette features and Cartesian component of optical flow velocity for action recognition. In this regard, human action models are created in each viewing direction for some specific actions using HMM. Ogale et al. [137] learn HMMs over cluster of key-frame silhouettes, which observe actions from different viewpoints.

The problem of noisy silhouette, e.g. in outdoor scenes where exact background segmentation is difficult, is tackled by using Chamfer distance [56] in [47, 197]. Robust matching of the noisy silhouettes is done by using phase correlation [137], by stacking the space time volume spanned by silhouette images over time [15, 63, 212], or by using shape context descriptors [121, 171, 219].

Contour/Pose based

Contours are used in [33], where the star skeleton extracted from the contour describes the angle between a reference line, and the lines from the center to the gross extremities (head, feet and hands) of the contour. In the work of Wang et al. [191] both silhouette and contour descriptors are used. Given a sequence of frames, an average silhouette is formed by calculating the mean intensity over all centered contours of all frames. Mendoza et al. [127] use contour histogram of full body and HMM for human action recognition. Likewise, Sundaresan et al. [177] use the sum of silhouette pixels. Full human body poses are also taken into consideration for action recognition. Wang et al. [192] propose a directional HOG feature [40] for pose estimation and action recognition is done using pose similarity measure. Rittscher et al. [153] recognize actions from a condensation filter which uses spline contours to model the evolution of a person outline over time.

Angular features

In the work of Ali et al. [5] angle subtended by torso and legs with the vertical axis is used to recognize actions. In this work, first background subtraction is applied to extract human. Skeletonization is used to detect hip and knee regions by estimating the highest points of the curvature on the skeleton. Angle subtended by torso and legs with the vertical axis are observed. In an action sequence those angles traverse a path of maxima and minima over time dimension. Euclidean distance has been used for classifying test sequences. Similar approach is also proposed by Chakraborty et al. [28] for real time human action recognition.

As demonstrated by many of the above mentioned approaches, spatial representation of the action using silhouettes and contour provide strong cues for action recognition, and moreover have the advantages of being insensitive to colour, texture and contrast changes. On the negative side, silhouette based representations fail in detecting self-occlusions and depend on robust background subtraction.

2.1.2 Temporal/motion representation:

In this category, motion based representation is used to model the temporal information of action. This is a clear intuition that motion model is one of the best cue to find underlying action characteristics.

Optical flow based

Instead of using shape information from the full body, motion information can also be used. An early example of using optical flow for action recognition is introduced by Polana et al. [147], where they compute *temporal-textures* i.e. first and second order statistics based on the direction and magnitude of normal flow, to recognize events such as motion of tress in wind or turbulent motion of water. In [146], Polana et al. propose features for human action recognition based on flow magnitudes accumulated in a regular grid of non-overlapping bins. Another early approach using optical flow is proposed by Cutler et al. [39] where the optical flow field is clustered into a set of *motion blobs*, and motion, size and position of those blobs are used as features for action recognition. More recently, Efros et al. [46] compute optical flow in person centered images. This algorithm is applied on sports footage, where persons in the images are very small. To avoid the cancellation effect of the oppositely directed vectors, the horizontal and vertical components are divided into positive and negative directions, yielding 4 distinct channels. Ahad et al. [3] use these four flow channels to solve the issue of self-occlusion in a MHI approach. Ali et al. [7] derive a number of Kinematic features from optical flow. These include divergence, voracity, symmetry and gradient tensor features. Principal component analysis (PCA) is applied to determine dominant kinematic modes. In the works [50, 91, 105], the adaboost-based Viola-Jones face detector [187] is extended to action recognition by replacing the rectangular image features with spatio-temporal cubes over optical flow.

A framework for modeling and recognition of temporal activities are proposed in [14]. Here an observed activity is represented as a vector of measurements over the temporal axis. The objective of this work is to develop a method for modeling and recognition of these temporal measurements while accounting for some of the above variances in activity execution. It uses optical flow based body part recognition and extracts motion parameters from different body parts like, head, torso, arms, legs etc to recognize actions.

Flow based representations do not depend on background subtraction, which makes them practical than silhouettes in many settings, because they do not require background models. On the downsides, they reply on the assumption that image differences can be explained as a result of movement, rather than changes in material properties, lighting etc.

Key-pose based

Instead of modeling motion or action dynamics, there are approaches attempt to recognize actions from isolated, characteristic *key-frames* or with other time-independent measures such as frequency of feature occurrence to capture the temporal evolution of action. Carlsson et al. [26] introduce the use of *key-frames*, i.e. a single characteristic frame of an action, to recognize forehand and backhand strokes in tennis recordings.

Instead of using single frame Schindler et al. [157] use a very short *snippets* of frames and try to answer the question of how many frames are required to perform action recognition? Pose contour based *key-poses* extraction is proposed by Baysal et al. [11], where nearest neighbour classifier is used for action recognition.

2.1.3 Global grid/part based:

The problem of view-point changes, noise and partial occlusions can partially be avoided by dividing the action region into a fixed spatial or temporal grid. In this case, each cell in the grid describes the image observation locally, and the matching function is changed accordingly from global to local.

Optical flow, gradient and local pattern based

Grid based techniques attempt to model the information of each grid segments by the optical flow, gradients or local patterns. Kellokumpu et al. [92, 93] calculate local binary patterns along the temporal dimension and store a histogram of non-background response in a spatial grid. Zelnik et al. [218] compute gradient fields in XYT direction and represent each frame through the histogram over those gradients. Thurau et al. [179] use histograms of oriented gradients (HOG, [40]) and focus on foreground edges by applying non-negative matrix factorization. Lu et al. [118] apply PCA after calculating the HOG descriptor, which greatly reduces the dimensionality. Ragheb et al. [150] transform, for each spatial location, the binary silhouette response over time into the frequency domain. Each cell in the spatial grid contains the mean frequency response of the spatial locations it contains. Optical flow in a grid-based representation is used in the work of Danafar et al. [41]. They adapt the work of Efros et al. [46] by dividing the ROI into horizontal slices that approximately contain head, body and legs. Zhang et al. [219] use an adaptation of the shape context, where each log-polar bin corresponds to a histogram of motion word frequencies. Combinations of flow and shape descriptors are also common, and overcome the limitations of a single representation. Tran et al. [180] use rectangular grids of silhouettes and flow. Within each cell, a circular grid is used to accumulate the responses. Ikizler et al. [77] combine the work of Efros et al. [46] with histograms of oriented line segments. Flow, in combination with local binary patterns is used in [208].

Body-parts based

Instead of applying a fixed grid, there are actions which can be better recognized by only considering body parts, such as the dynamics of the legs for walking, running and jogging [42]. This work presents an approach for recognizing human walking movements using low-level motion regularities and constraints. The features for classification are automatically extracted from walking video sequences. The person is first tracked from video using [69]. Silhouette extraction and identification of head, torso and leg are then performed. Dynamic Regularity Features like cycle time, swing-stance ratio and double-support time are then collected from those body parts to detect the walking action.

Other work of action recognition based on a prior detection of the human body parts is by Park et al. [143]. This work describes a framework for recognizing human actions and interactions in video by using three levels of abstraction. At low level, the poses of individual body parts including head, torso, arms and legs are recognized using individual Bayesian networks (BNs), which are then integrated to obtain an overall body pose. At mid level, the actions of a single person are modeled using a dynamic Bayesian network (DBN) with temporal links between identical states of the Bayesian network at time t and $(t + 1)$. At high level, the results of mid-level descriptions for each person are combined together along a common time line to identify an interaction between two persons. Mori et al. [132, 131] utilize the geometrical models of human body parts where action is recognized by searching for the static postures in the image that match the target action. Recent works on part based event detection use hidden conditional random field [195, 194], flow-based shape feature [91] and histogram of oriented rectangles [78].

2.1.4 Contributions to the state-of-the-art

In Chapter 3, we present a body-part based human action recognition, where stochastic movements of the ensemble of the body-parts are modeled using HMM. This strong model based approach is able to recognize actions by only considering the body-parts having major contribution to the actions, for example, legs for the walking, jogging and arms from boxing, clapping. The HMM construction uses an ensemble of body-part detectors, followed by grouping of part detections, to perform human identification.

In Chapter 4, a weak model based human action recognition is presented, where actions are represented by using Bag-of-key poses model. The key-poses are obtained using K-medoids clustering on the limb dynamics features. Limb dynamics are modeled using the directional HOG features. For limb region detection, we introduce a scale adaptive ASEF human detector, where first an ASEF human detector is used to identify initial hypothesis for a human and a HOG + SVM human detector is applied to locate the human more robustly.

2.2 Local representation

Local representations describe the observation as a collection of local descriptors or patches. We term this representation a *model free approach* as the local patches are not following any strict model. Accurate localization and background subtraction are not required in this case and local representation are somewhat invariant to changes in viewpoint, person appearance and and partial occlusions.

2.2.1 Spatio-temporal interest points (STIPs) based

The extraction of appropriate features is critical to action recognition. Ideally, visual features are able to handle the following challenges for robust performance: (i) scale, rotation and viewpoint variations of the camera, (ii) performance speed variations for different people, (iii) different anthropometry of the actors and their movement style variations, and (iv) cluttered backgrounds and camera motion. The ultimate goal is to be able to perform reliable action recognition applicable for video indexing and search, intelligent human computer interaction, video surveillance, automatic activity analysis and behavior understanding. Recently, the use of STIPs has received increasing interest for local descriptor-based action recognition strategies. STIP-based methods avoid the temporal alignment problem, are exceptionally invariant to geometric transformations, and therefore distorted less by changes in scale, rotation and viewpoint than image data. Features are locally detected, thus inherently robust to occlusion and do not suffer from conventional figure-ground segmentation problems (imprecise segmentation, object splitting and merging etc.). Additionally, partial robustness to illumination variations and background clutter are incorporated.

Laptev and Lindeberg first proposed STIPs for action recognition [102], by introducing a space-time extension of the popular Harris detector [70]. They detect regions having high intensity variation in both space and time as spatio-temporal corners. The STIP detector of [102] usually suffers from sparse STIP detection. Later several other methods for detecting STIPs have been reported [44, 80, 139, 202, 203]. Dollar et al. [44] improved the sparse STIP detector by applying temporal Gabor filters and select regions of high responses. Dense and scale-invariant spatio-temporal interest points were proposed by Willems et al. [202], as a spatio-temporal extension of the Hessian saliency measure, previously applied for object detection [12, 110]. Instead of applying local information for STIP detection Wong et al. [203] propose a global information-based approach. They use global structural information of moving points and select STIPs according to their probability of belonging to the relevant motion. Although promising results have been reported, these methods are quite vulnerable to camera motion and cluttered background, since they detect interest points directly in a spatio-temporal space.

Hence, STIP-based methods have some shortcomings. First of all, (i) STIPs focus on local spatio-temporal information instead of global motion, thus the detection of STIPs on human actors in complex scenes might fall on cluttered backgrounds, especially if the camera is not fixed. Secondly, (ii) the stability of STIPs varies due to the local properties of the detector, and therefore some STIPs can be unstable and imprecise, as a result they have low repeatability or the local descriptors can become ambiguous. Thirdly, (iii) redundancy can occur in the local descriptors extracted from the surrounding image region of two adjacent STIPs. According to Schmid et al. [158] robust interest points should have high repeatability (geometric stability) and information content (distinctiveness of features). Furthermore, Turcot et al. [182] investigate and report that it is better to select a small subset of useful features for recognition problems, than a larger set of unreliable features which represent irrelevant clutter.

Applying these STIP detectors in complex action datasets, like in [113, 125], results in a large number of background STIP detection. Those unwanted background STIPs usually gives erroneous the action recognition. To overcome this problem, two main directions have been followed. Methods like [19, 58, 203] apply different way of STIP computation from [102]. Wong et al. [203] propose a global structural information based approach for the STIP computation. Bregonzio et al. [19] modify the Dollar STIP detector [44] by applying $2D$-Gabor filter. These methods work on relatively simple datasets and are not robust enough for the complex datasets with unconstrained environment. Gilbert et al. [58] use very dense corner features that are spatially and temporally grouped in a hierarchical process to produce an over complete compound feature set. This may introduce a large amount of noise and consequently, results in a weak feature extraction.

Other approaches [20, 115, 184] first apply the STIP detector of [44, 102] and then use different heuristics to prune the detected STIPs. Liu et al. [115] use static and motion feature pruning using different rule based heuristics, ROI estimation and page ranking. Bregonzio et al. [20] compute trajectories of the detected STIPs and use KLT tracker and feature selection to prune the unwanted STIPs. Ullah et al. [184] use video segmentation prior to apply the STIP detector to get rid of the unwanted STIPs. The main drawback of these methods is the application of different heuristic based pruning or many complex pre/post processing like, segmentation, human model and tracking. Even with the addition of all these complex processing steps, significantly high recognition performance has yet not been achieved for complex datasets.

Several local descriptors have been proposed in the past few years [44, 202, 95, 96, 103, 104, 160]. Local feature descriptors extract shape and motion in the neighborhoods of selected STIPs using image measurements, such as spatial or spatio-temporal image gradients or optical flow. Laptev et al. [104] introduced a combined descriptor to characterize local motion and appearance by computing histograms of spatial gradient (HOG) and optic flow (HOF) accumulated in space-time neighborhoods of detected interest points. Willems et al. [202] proposed the Extended SURF (ESURF) descriptor, which extends the image SURF descriptor [10] to videos. The authors divide 3D patches into cells, where each cell is represented by a vector of weighted sums of uniformly sampled responses of the Haar-wavelets along the three axes. Dollar et al. [44] proposed a descriptor along with their detector. The authors concatenate the gradients computed for each pixel in the neighborhood into a single vector and apply Principal Component Analysis (PCA) to project the feature vector onto a low dimensional space. Compared to the HOG-HOF descriptor proposed by Laptev et al. [104], it does not distinguish the appearance and motion features. The 3D-SIFT descriptor was developed by Scovanner et al. [160]. This descriptor is similar to the Scale Invariant Feature Transformation (SIFT) descriptor [117], except that it is extended to video sequences by computing the gradient direction for each pixel spatio-temporally in three-dimensions. Another extension of the popular SIFT descriptor was proposed by Kläser et al. [95]. It is based on histograms of 3D gradient orientations, where gradients are computed using an integral video representation. Another popular descriptor is the N-jets [96, 101]. An N-jet is the set of partial derivatives of a function up to order N, and is usually computed from a scale-space representation. The N-jets is an inherently strong local motion descriptor, where the two first levels implicitly represent velocity and acceleration.

Bag-of-video words (BoV) models have become popular for generic action recognition [44, 102, 115, 113, 203, 217], whereas other techniques based on co-occurrence of STIP based motion features are also used [126]. The basic BoV model computes and quantizes the feature vectors, extracted at the detected STIPs in the video, into video-words. Finally, the entire video sequence is represented by a statistical distribution of those video-words. For classification, discriminative learning models such as SVM [44] and generative models, e.g. pLSA [203], have achieved excellent performance for action recognition. Since the BoV model does not provide a spatio-temporal distribution of features, the spatial correlogram and spatio-temporal pyramid matching are applied [113, 125] to capture the spatio-temporal relationship between local features. Additionally, vocabulary compression techniques are used to reduce the final feature space [115, 113].

2.2.2 Space-time trajectories based

Trajectory based approaches are recognition approaches that interpret an activity as a set of space-time trajectories. Trajectories are generally obtained from the 2D (XY) or 3D (XYZ) points on the human corresponding to human body joint positions or STIP points.

The early work on this category is done by Johansson et al. [84], where it is shown that tracking of joint positions itself is sufficient for humans to distinguish actions, and this paradigm is further studied for the human activity recognition [196, 136].

Several approaches used the trajectories themselves (i.e. set of 3D points) to represent and recognize actions directly [162, 213]. Sheikh et al. [162] represent an action as a set of 13 joint trajectories in a 4D ($XYZT$) space. An affine projection is used to obtain normalized XYT trajectories of an action, to measure view-invariant similarity between two sets of trajectories. Yilmaz et al. [213] present a methodology to compare action videos obtained from moving cameras, also using a set of 4D $XYZT$ joint trajectories.

Recent work of trajectory based action recognition is motivated by the dense point sampling in a video. Wang et al. [189] use dense points from each frame and track them based on displacement information from a dense optical flow field. Sun et al. [175] propose a scheme of extraction and representation of dense, long-duration trajectories from video sequences, and demonstrate its ability to handle video sequences containing occlusions, camera motions, and nonrigid deformations.

2.2.3 Contributions to state-of-the-art

In Chapter 5 a novel STIP detector is described which is different from [44, 102] by extending the 2D Harris corner detector. We first detect Spatial Interest Points (SIPs), then suppress unwanted background SIPs, and finally impose local and temporal constraints, achieving a robust set of STIPs. The suppression mask uses a mechanism similar to that of the non-classical receptive field inhibition that is exhibited by most orientation selective neurons in the primary visual cortex. This concept is motivated by the work of Grigorescu et al. [65], where surround suppression is used on texture edges to improve object contour and boundary detection in natural scenes. Our method avoids complex processing, like in [20, 115, 184] yet improves the performance. Moreover, separating spatial and temporal constraint of STIPs uncovers static and dynamic characteristics of the motion by detecting STIPs on the static regions of the actors. This is an important factor to boost up the overall recognition rate since actions are best describe by the combination of moving and non-moving parts of the actor.

For action representation a novel BoV model is introduced using a compact and efficient pyramid representation by applying a spatial pyramid at the STIP domain to group the local motion features, together with a vocabulary compression at each pyramid level. This is different from [113], where first a vocabulary is computed, then it is compressed, and finally a spatial correlogram and a spatio-temporal pyramid are applied.

2.3 Action recognition in multi-camera environment

A 3D data representation is more informative than the analysis of 2D activities carried out in the image plane, which is only a projection of the actual actions. As a result, the projection of the actions will depend on the viewpoint, and not contain full information about the performed activities. To overcome this shortcoming the use of 3D data has been introduced through the use of two or more cameras [35, 60, 167, 200]. In this way the surface structure or a 3D volume of the person can be reconstructed, e.g., by Shape-From-Silhouette (SFS) techniques [173], and thereby a more descriptive representation for action recognition can be established.

The use of 3D data allows for efficient analysis of 3D human activities. However, we are still faced with the problem that the orientation of the subject in the 3D space should be known. Therefore, approaches have been proposed without this assumption by introducing view-invariant or view-independent representations.

2.3.1 View-Invariant 2D Feature Description

One line of work concentrates solely on the 2D image data acquired by multiple cameras [61, 79, 81, 172]. In the work of Souvenir et al [172] actions are described in a view-invariant manner by computing $\mathcal{R}$ transform surfaces of silhouettes and manifold learning. Gkalelis et al. [61] exploit the circular shift invariance property of the discrete Fourier Transform (DFT) magnitudes, and use Fuzzy Vector Quantization (FVQ) and Linear Discriminant Analysis (LDA) to represent and classify actions. Another approach was proposed by Iosifidis et al. [79], where binary body masks from frames of a multi-camera setup used to produce the i3DPost Multi-View Human Action Dataset [60], are concatenated to multi-view binary masks.

Some authors perform action recognition from image sequences in different viewing angles. Ahmad et al. [4] apply Principal Component Analysis (PCA) of optical flow velocity and human body shape information, and then represent each action using a set of multi-dimensional discrete Hidden Markov Models (HMM) for each action and viewpoint. Cherla et al. [34] show how view-invariant recognition can be performed by using data fusion of two orthogonal views. An action basis is built using Eigen analysis of walking sequences of different people, and projections of the width profile of the actor and spatio-temporal features are applied. Finally, Dynamic Time Warping (DTW) is used for recognition. A number of other techniques have been employed, like metric learning [180] or representing action by feature-trees [152] or ballistic dynamics [188]. In [199], Weinland et al. propose an approach which is robust to occlusions and viewpoint changes using local partitioning and hierarchical classification of 3D Histogram of Oriented Gradients (3DHOG) volumes.

Others use synthetic data rendered from a wide range of viewpoints to train their model and then classify actions in a single view, e.g. Lv et al. [121], where shape context is applied to represent key poses from silhouettes and Viterbi Path Searching for classification. A similar approach was proposed by Fihl et al. [51] for gait analysis.

Another topic which has been explored by several authors the last couple of years is cross-view action recognition. This is a difficult task of recognizing actions by training on one view and testing on another completely different view (e.g., the side view versus the top view of a person in IXMAS). A number of techniques have been proposed, stretching from applying multiple features [111], information maximization [113], dynamic scene geometry [68], self similarities [86, 87] and transfer learning [49, 114]. For additional related work on view-invariant approaches please refer to the recent survey by Ji et al. [81].

2.3.2 3D Shape and Pose Descriptors

Another line of work utilize the full reconstructed 3D data for feature extraction and description ([8, 85, 90, 97, 142]). Johnson and Hebert proposed the spin image [85], and Osada et al. the shape distribution [142]. Ankerst et al. introduced the shape histogram [8], which is a similar to the 3D extended shape context [13] presented by Körtgen et al. [97], and Kazhdan et al. applied spherical harmonics to represent the shape histogram in a view-invariant manner [90]. Later Huang et al. extended the shape histogram with color information [74]. Recently, Huang et al. made a comparison of these shape descriptors combined with self similarities, with the shape histogram (3D shape context) as the top performing descriptor [75, 76].

A common characteristic of all these approaches is that they are solely based on static features, like shape and pose description, while the most popular and best performing 2D image descriptors apply motion information or a combination of the two [129, 104, 115, 201].

2.3.3 3D Motion Descriptors

Instead of only relying on static features, some authors add temporal information by capturing the change of static descriptors over time, i.e., shape and pose changes,by accumulating static descriptors over time, track human shape or pose information, or apply sliding windows to capture the temporal contents [129, 144, 145, 200, 198]. Cohen et al. [35] use 3D human body shapes for view-invariant identification of human body postures, and compute an invariant measure of the distribution. They apply a cylindrical histogram and compute an invariant measure of the distribution of reconstructed voxels, which later was used by Pierobon et al. [145] for human action recognition.

The Motion History Volume (MVH) was proposed by Weinland et al. [200], as a 3D extension of Motion History Images (MHIs) [16]. MHVs are created by accumulating static human postures over time in a cylindrical representation, which is made view-invariant with respect to the vertical axis by applying the Fourier transform in cylindrical coordinates. The same representation was used by Turaga et al. [181] in combination with a more sophisticated action learning and classification based on Stiefel and Grassmann manifolds. Later, Weinland et al. [198] proposed a framework, where actions are modeled using 3D occupancy grids, built from multiple viewpoints, in an exemplar-based Hidden Markov Models (HMM). Learned 3D exemplars are used to produce $2D$ image information which is compared to the observations, hence, 3D reconstruction is not required during the recognition phase.

Pehlivan et al. [144] presented a view-independent representation based on human poses. The volume of the human body is first divided into a sequence of horizontal layers, and then the intersections of the body segments with each layer are coded with enclosing circles. The circular features in all layers: (i) the number of circles, (ii) the area of the outer circle, and (iii) the area of the inner circle are then used to generate a pose descriptor. The pose descriptors of all frames in an action sequence are further combined to generate corresponding motion descriptors. Action recognition is then performed with a simple nearest neighbor classifier. Canton-Ferrer et al. [24] propose another view-invariant representation based on 3D MHIs and 3D invariant statistical moments [116]. A different strategy is presented by Yan et al. [207]. They propose a 4D action feature model (4D-AFM) for recognizing actions from arbitrary views based on spatio-temporal features of spatio-temporal volumes (STVs) [212]. The extracted features are mapped from the STVs to a sequence of reconstructed 3D visual hulls over time, resulting in the 4D-AFM model, which is used for matching actions.

Another pair of 3D descriptors which are based on rich motion information are the 3D Motion Context (3D-MC) and the Harmonic Motion Context (HMC) proposed by Holte et al. [72] The 3D-MC descriptor is a motion oriented 3D version of the shape context [13, 97], which incorporates motion information implicitly from 3D optical flow. The HMC descriptor is an extended version of the 3D-MC descriptor that makes it view-invariant by decomposing the representation into a set of spherical harmonic basis functions.

2.3.4 Contributions to state-of-the-art

In the Chapter 6, we present a novel approach for human action recognition in multi-camera environment. In this work we stress the application of model free approach, STIP and local feature based approaches, instead of model based approaches which are commonly applied by most state-of-the-art methods. We extend the selective STIP detector of [27], to a novel 4D STIPs by combining the STIPs from each view-point by using the reconstructed 3D data acquired by multi-camera systems. We also introduce a novel view-invariant 3D motion descriptor, Histogram of Optical 3D Flow for action representation. Opposed to other for 3D action recognition, which are solely based on holistic features, e.g. [72, 144, 172, 200], our approach extends the concepts of STIP detection and local feature description for building a Bag-of-Words (BoW) vocabulary of human actions, which has gained popularity in the 2D image domain, to the 3D case.

2.4 Activity detection

Activity detection is a part of the action recognition where not only actions are to be recognized but also the location, start and end frame of the actions should be detected. This is relative newly emerged area of action recognition and mainly use the small scale datasets. Very recently concept of continuous visual event recognition (CVER) in large scale has been introduced.

Methods in this category use the techniques for single camera based action recognition described in the Section 2.1 for identifying the action region and for action representation. In this regard, STIPs and local features based method is a popular choice.

2.4.1 Activity detection on small scale dataset:

In the activity detection important thing is to locate the action, and exhaustive search is practically infeasible due the enormous volume of the search space. To tackle this issue, Yuan et al. [217] propose a sub-volume search using branch and bound method proposed by Lampert et al. [99] in the feature space, where the entire video is presented by STIP and local motion features. Hu et al. [73] employ multiple instance learning framework for rough annotation. Cao et al. [25] propose Gaussian mixture model based action detection. In this approach cross dataset evaluation is proposed, where the training and test dataset are different. Random forest based indexing is proposed by Yu et al. [215] for action search.

Having inspired by the success of the Hough transform-based methods in object detection [52, 106, 107, 123, 140, 141], methods like [53, 211] use Hough transform-based voting framework for action classification and localization. In this approach dense features (See Section 2.2) are first computed and then a Hough forest is built to train the actions. Although promising but these works always use small scale video datasets like KTH [159] and MSR action datasets [217].

2.4.2 Activity detection on large scale dataset:

In general, research towards large scale action detection from video is less explored. Few works like [138, 25] shows some progress in this area. To solve the search space complexity of CVER in large scale, Oh et al. [138] apply a multi-object tracking using frame difference, and the obtained tracks are divided into detection units resulting over $20K$ units, as a preprocessing step. A BoV + SVM like [159] is used for activity recognition. Other work on the large scale activity recognition is by Cao et al. [25], where the result on TRECVID 2008 dataset [43] is presented, although only one action type, "running", is used.

Table 2.4.3 describes the different methods for action recognition and activity detection using the taxonomy defined in this chapter. We also include the methods that are presented in this thesis work.

2.4.3 Contribution to the state-of-the-art:

In the chapter 7, we present a novel method for activity recognition in large scale dataset. In our approach first a motion segmentation method similar to [174] is applied to obtain the primary candidate region set. The obtained regions are further joined using a region clustering technique based on action heuristics. Finally, we obtain on an average $3K$ candidate regions as opposed of $20K$ as in [138] with a greater recall rate. This has a major impact towards reducing the search space and on achieving faster event detection in large scale.

For activity detection, we extend the max-margin Hough transformation technique of [123] used in 2D object recognition to activity recognition. We perform test on large scale video dataset of [138] as well as small scale dataset MSR [217] and obtain state-of-the-art result.

Visual Event Search												
Referances	Single Cam. (2*D*)								Multi-view (3*D*)			Type
	Full Body						Local		Shape		Motion	
	Spatial			Motion/Temp.		Grid	STIP	Trj.	2D	3D		
	Sil.	Cont.	Ang.	Opt. flow	Key-pose							
[16, 66, 136, 172, 193, 198]	✓											R
[33, 127, 153, 191, 192]		✓										R
[5, 28]			✓									R
[3, 7, 14, 39, 46, 147, 146]				✓								R
[11, 26, 157]					✓							R
[42, 69, 143, 132, 195, 78]						✓						R
[19, 44, 58, 80, 102, 113, 125, 139, 202, 203, 184]				✓		✓	✓					R
[84, 162, 175, 189, 213]						✓		✓				R
[34, 51, 61, 79, 81, 121, 152, 172, 180, 188]									✓			R
[111, 113, 68, 86, 87, 49, 114, 81]										✓		R
[24, 35, 72, 144, 145, 129, 181, 200, 198]											✓	R
[217, 99, 25, 215]				✓		✓	✓					AD
[138, 25, 43]				✓		✓	✓					LAD

Table 2.1: Overview of the state-of-the-methods based on our taxonomy on the visual event recognition. The abbreviations, "R" denotes **action recognition**,"AD" denotes **activity detection** and "LAD" denotes **activity detection in large scale.**

Chapter 3

Human action recognition using ensemble of body-parts

> *"Divide each difficulty into as many parts as is feasible and necessary to resolve it."*
> *Le Discours de la Méthode* (1637), by René Descartes

Recognizing human action is primarily a task of analyzing the movement of body-parts and often these movements are stochastic in nature. This chapter explores this idea and describes an approach to human action recognition based on a probabilistic optimization model of body parts using Hidden Markov Model (HMM). This method is able to distinguish between similar actions by only considering the body parts having major contribution to the actions, for example, legs for walking, jogging and running; arms for boxing, waving and clapping. The HMM construction uses an ensemble of body-part detectors, followed by grouping of part detections, to perform human identification. Hence this chapter establishes the idea of using a strong model e.g. body-part detector to identify the regions which are important for action recognition.

Understanding human actions is to analyze the individual movement of different participating body-parts. In most of the basic human actions like walking, running, boxing and waving etc these movements are repeating in nature. When viewed as a stochastic estimation problem, the critical issue in action recognition becomes the definition and computation of the likelihood, $\Pr(a|H, I)$, of action a given the human H in the image sequence I. The difficulty in working with this likelihood is directly related to the complexity of the joint distribution $\Pr(H, I)$ over all possible human figures H in the image sequence I. Holistic approaches which attempt to model the entire human figure, generally, must resort to very sophisticated and complex models of this joint distribution, resulting in very demanding model estimation and optimization problems.

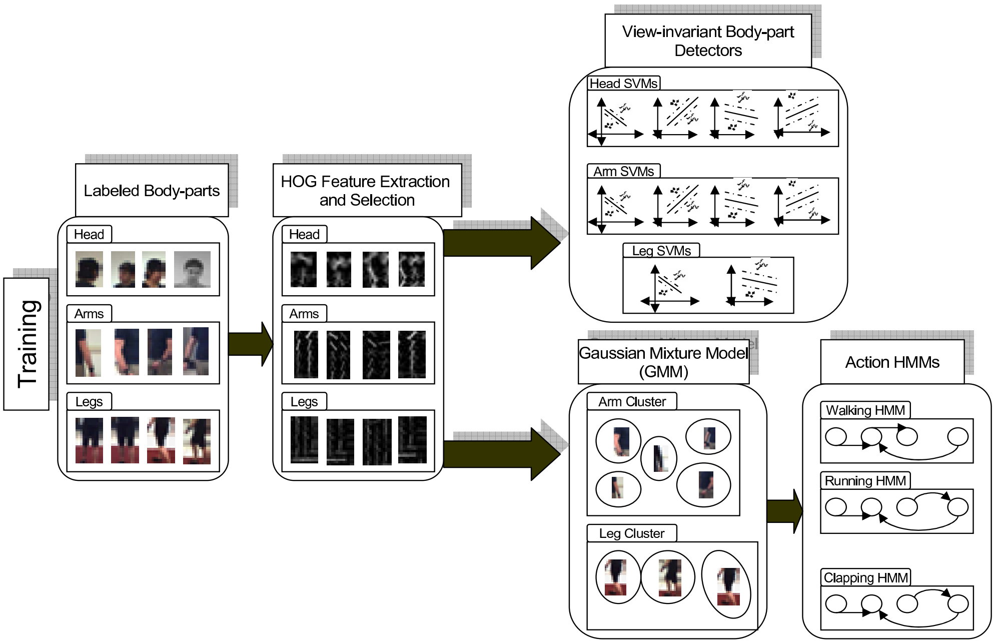


Figure 3.1: The learning process for body-part detectors and action HMMs. Labeled body part images are fed into a feature extraction and selection module. These features are then sent, in parallel, to an SVM learning phase which trains viewpoint-invariant body part detectors and to a GMM fitting phase that finds the key-poses of body parts. Sequences of these body part key-poses are then used to learn an action HMM for each class. Classification is performed by applying the body-part detectors to video sequences, applying a geometric constraint model to filter out false-positive detections and finally by estimating the MAP-likelihood for each trained action HMM to generate the sequence of detected key-poses.

As discussed before, basic human actions can be represented using the local motion of individual body parts. Actions like walking, jogging, running, boxing and waving are systematic combinations of the motion of different human body components. From this perspective, it can also be observed that not all body parts contribute equally to all action classes. For example, actions like walking, running and jogging are characterized mostly by the movement of the legs. Boxing, waving and hand clapping, on the other hand, mostly depend on the arms.

Based on these observations, we define the action likelihood $\Pr(a|H, I)$ instead as $\Pr(a|B, I)$, where B is an ensemble of body parts which the human H is composed of. Moreover, the likelihood is further simplified by conditioning actions only on those body parts which contribute most to a particular action. Features from body parts B are used to model the action likelihood $\Pr(a|B, I)$, and optimizing this likelihood over all known actions yields a maximum likelihood estimate of the action in the image sequence.

The ensemble of body part detectors is an important component of our method and we use SVM-based body part detectors over a range of viewpoints to build viewpoint-invariant body part detectors. We model human actions as a repeating chain of body-part poses [30]. Similar poses are grouped together by applying a Gaussian Mixture Model (GMM) on the features of the body parts in order to identify key-poses. These key-poses then serve as a vocabulary of hidden action-states for Hidden Markov Models (HMMs) that model the temporal-stochastic evolution of each action class. Figure 3.1 shows an overview of the proposed method.

3.1 A probabilistic model for action recognition

The task of human action recognition can be formulated as an optimization problem. Let $A = \{a_1, a_2, \ldots, a_n\}$ denote a set of possible actions, where each a_i denotes a specific action such as walking, running or hand clapping. We write the likelihood of a specific action a_i in a given image sequence I with human H as:

$$\Pr(a_i|H, I) \text{ for } a_i \in A.$$

Given an instance of I with detected human H, a maximum likelihood estimation of the action being performed is:

$$a^* = \operatorname*{argmax}_{a_i \in A} \Pr(a_i|H, I). \tag{3.1}$$

Rather than holistically modelling the entire human H, we consider it to be an ensemble of detectable body parts:

$$B = \{B_1, B_2, \ldots, B_m\},$$

where each B_i represents one of m-body parts such as the head, legs, arms or torso. The likelihood can now be expressed as:

$$\Pr(a_i|H, I) = \Pr(a_i|\{B_1, B_2, ..., B_m\}, I) \tag{3.2}$$

Our model is based on the fact that not all actions depend equally on all body parts. Actions like walking, jogging and running, for example, depend primarily on the legs. Actions like boxing, waving and clapping, on the other hand, depend primarily on the arms. To simplify the likelihood in (Equation 3.2), we define a dependence function over the set of all subsets of body parts:

$$d(a_i) : A \longrightarrow \mathcal{P}(B),$$

where $\mathcal{P}(B)$ is the power set of B, or $\{c|c \subseteq B\}$.

The set $d(a_i)$ determines the subset of body parts on which action a_i most strongly depends. The likelihood is now approximated as:

$$\Pr(a_i|H, I) \approx \Pr(a_i|d(a_i), I), \tag{3.3}$$

and the maximum likelihood estimate of the action given an ensemble of body parts and an image sequence is:

$$a^* = \operatorname*{argmax}_{a_i \in A} \Pr(a_i | d(a_i), I). \tag{3.4}$$

The approximate likelihood in Equation 3.3 makes explicit the dependence of an action class on a *subset* of body parts. This approximation assumes that the action a_i is independent of the body parts excluded from $d(a_i)$ and thus the likelihood can be computed by simply excluding the irrelevant parts rather than optimizing (or integrating) over them. In the following sections we describe how we model the approximate likelihood in Equation 3.3 using viewpoint invariant body-part detectors and HMMs over detected features of these body parts, and then from these arrive at the estimate of Equation 3.4 for action classification.

3.2 Body-part detection

To obtain the $d(a_i)$ in Equation 3.3, we apply body-part based human detection. Here, we use three body part detectors for the head, leg and arms respectively. Our body part detection is based on *sliding-window* technique. Specific sized rectangular bounding boxes are used for each body part. These bounding boxes are slided over the image-frame, taking each sub-window considered as an input to the head, leg and arm detectors. These inputs are then independently classified as either a respective body part or a non-body part. In order to prevent possible false positives, those detected components are combined into a proper geometrical configuration into another specific sized bounding box as a full human. The sizes of all these bounding boxes are obtained from the training samples. Furthermore, the image itself is processed at several scales, yielding scale-invariant body-part detection.

3.2.1 Feature extraction and selection

In our approach (Algorithm 1), labeled training images (see Figure 3.5) for each body part detector are divided into (8×8) cells after applying a Sobel mask to them. HOG features are extracted from those cells and a normalized 6-bin histogram is computed over the angle range (-$\frac{\pi}{2}$, $+\frac{\pi}{2}$). So, this gives one $6D$ feature vector for each of those (8×8) pixel cells. Next, we select the best feature vector group among all of them. This feature selection method is based on the minimization of the standard deviation (σ) of different dimensions of those feature vectors. Note that our features are not precisely the HOGs defined in [40]. We do not compute the larger, overlapping and normalized blocks as in the original HOG, but rather limit our features to (8×8) cells upon which the final HOGs of Dalal and Triggs are based. The larger block normalization, in [40], is used for improving the robustness of the method to different illumination or shadow condition. But in the current work we have restricted the block normalization into cells. The datasets we have used in this work are mostly having controlled environments without wide illumination and shadow conditions. For this we keep our HOG implementation simple, restricting the block normalization to cells. Besides omitting larger block normalization results in low memory usage for the final HOG features. As our system works using different body-part detectors in parallel, this low memory usage for features increases the overall efficiency. Moreover, we omit this part in order to provide robustness to the final part detectors under the hypothesis that using many small features will be more resilient to missing ones due to occlusions or changing view-point. But larger block normalization may tackle more diverse and wide cases of illumination and shadow condition.

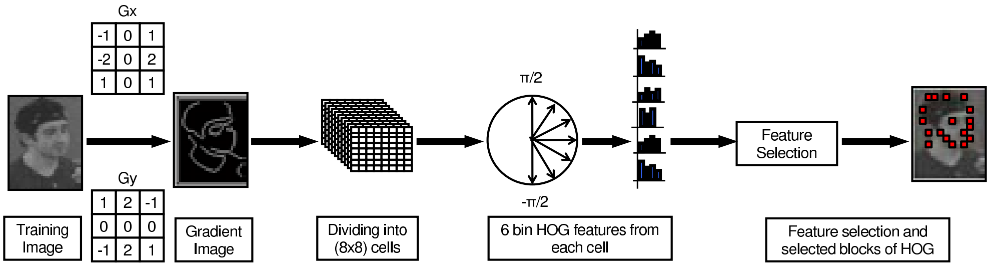

Gx
-1 0 1
-2 0 2
1 0 1
Gy
1 2 -1
0 0 0
-1 2 1
π/2
-π/2
Feature Selection
Training Image
Gradient Image
Dividing into (8x8) cells
6 bin HOG features from each cell
Feature selection and selected blocks of HOG

Figure 3.2: Feature extraction and selection method from a training image. The training image is divided into several cells and then HOG features are extracted. Finally a standard deviation based feature selection method is applied to obtain feature vector for SVM.

Algorithm 1 Feature extraction for body component SVM

```
Require: Training images of body component.
Ensure: Features for SVM.
for every training image do
    Apply Sobel operator
    Divide Sobel image into $(8 \times 8)$ cells
    for each cells do
        Compute normalized 6 bin histogram of HOG features within the range $(-\frac{\pi}{2}, +\frac{\pi}{2})$
        Keep it in feature array.
    end for
    Apply *feature selection* algorithm over the feature array.
end for
Selected features are used to learn SVM.
```

Let there be N training images of size $(W \times H)$, divided into n (8×8) cells. For each of these (8×8) cells we have a normalized 6-bin HOG feature vector, $G = \langle g_1, g_2, \ldots, g_6 \rangle$. The standard deviation, σ_{ij}, of i-th cell and j-th bin of HOG features:

$$\sigma_{ij} = \left(\frac{1}{N}\right) \times \sum_{t=1}^{N} (g_{ij}^{(t)} - \mu_{ij})^2 \tag{3.5}$$

where $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, 6$; $t = 1, 2, \ldots, N$; $g_{ij}^{(t)}$ is defined as j-th bin HOG feature of i-th cell of t-th training image and μ_{ij} is defined as the *mean* of j-th gradient of i-th cell over all the training images. The values of σ_{ij} are sorted and those $6D$ feature vector packets are taken for which σ_{ij} is smaller than a predefined threshold, ϵ. In our case we run the experiment several times to obtain this threshold empirically. These selected features are used to train SVMs for the body part detectors. Figure 3.2 shows the feature extraction technique.

To show the effectiveness of the proposed feature selection method, we perform an experiment to compare detection rates over a range of feature dimensionalities, both for our feature selection approach and for Principal Component Analysis (PCA) based feature extraction. We take different feature size by varying the fraction of original feature dimensionality included in the reduced representation and observe the recognition rate of the body-part detectors. Figure 3.3 shows a plot of recognition rate as a function of varying feature size. This graph clearly shows that the proposed feature selection method improves the recognition rate of the body part detectors compared to PCA-based approach over the same HOG features described above. It should also be noted that, for PCA-based feature extraction, all of the original features must be projected onto each principal eigen-vector. But, our approach avoids this need by directly using selected features from the original space, resulting in faster feature computation.

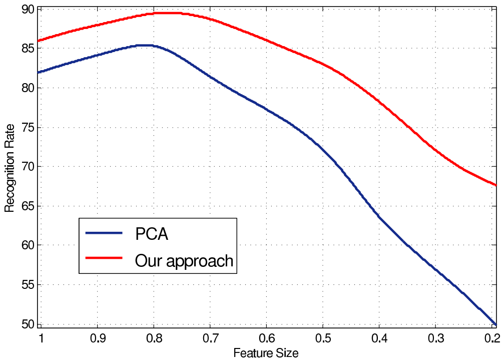


Figure 3.3: Feature dimensionality versus recognition rate. The graph shows body-part recognition rate for the head as a function of dimensionality with respect to the original feature dimensionality. X-axis shows the fraction of the original features taken after applying the feature selection process. It is clear from this experiment that our method of feature selection outperforms PCA-based feature extraction. These results are obtained from the head detector in the HumanEva dataset. This figure is best viewed in colour.

3.2.2 View-point invariance and partial occlusion

Part-based human detection can be sensitive to partial occlusions (see Figure 3.4). Given this, one must ask whether body-part detection works in all view-points and

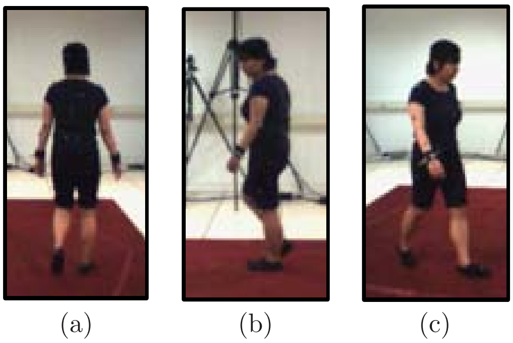


Figure 3.4: Partial occlusions in the walking action. (a) No occlusion, body parts: head, legs and arms, are visible. Partial occlusions, (b) one arm and one leg are occluded and (c) only one arm is visible.

how it handles partial occlusions of body parts. To tackle this problem, we design sub-classifiers within each of the three body-part detectors. Each of these sub-classifiers is designed to handle a range of view-points. Our system uses four head sub-detectors, two leg sub-detectors and four arm sub-detectors. The four head detectors correspond to view angle ranges $(\frac{\pi}{4}, \frac{3\pi}{4})$, $(\frac{3\pi}{4}, \frac{5\pi}{4})$, $(\frac{5\pi}{4}, \frac{7\pi}{4})$ and $(\frac{7\pi}{4}, \frac{\pi}{4})$. We chose this division so that the consecutive ranges have smooth transition of head poses. For arms, there are four classifiers corresponding to different arm positions, grouped in the same angular views of the head. Detecting arms is a difficult task since they have more degrees of freedom compared to the other body parts. For each action we use four major arm poses and considering the pose symmetry the detection of other pose variation is achieved. To detect the legs, two sub-classifiers have been used: one representing front (rear) view and the other one for profile views of the legs. Figure 3.5 shows some training samples from our body-part dataset.

We use the HumanEva dataset to create our body-part dataset and to train all body-part (sub-) detectors. The HumanEva dataset contains videos from 7 different view-points, as well as actions with circular patterns which result in many partial occlusions of body-parts. By using these view-point based examples to design different sub-classifiers allows us to handle partial occlusion. In each frame we apply these sub-classifiers and choose those with higher recognition scores. In this way, we obtain view-point invariant body part detection.

It is also worth noting, as will become more clear after defining the action HMM model in Section 3.3, that our final action classifier is also robust to detection failures. That is, it is not dependent on perfect human detection in every frame, but rather just on detection of enough representative body-part configurations (key-poses) to characterize an action.

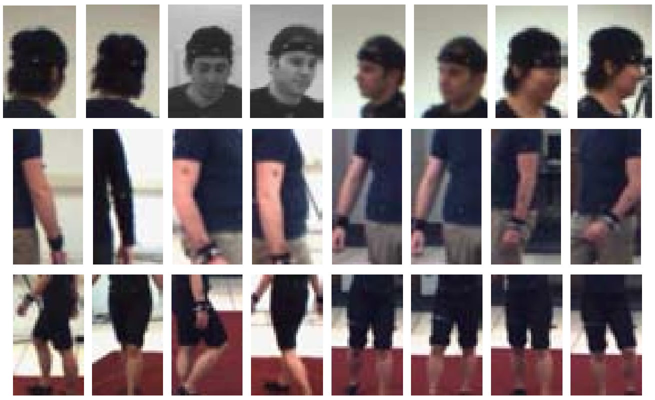


Figure 3.5: Training dataset for each body-part detectors. It shows training samples for each sub-classifier.

3.2.3 Geometric constraints on body-part detection

The outputs of the component detectors described above usually give a number of false positives (see Figure 3.8(b)) along with correctly detected body parts. We must combine these detector outputs in order to achieve full human detection and to eliminate false positives. The methods for the body part combination either use a probabilistic approach [204, 108, 128] or apply some non-probabilistic, rule based technique [163, 151, 130]. Probabilistic approaches compute the likelihood of the presence of multiple humans in the hypothesized region. The human model is constructed using the detected body parts. The part combination that gives the maximum likelihood is the region of detected human. This method is quite popular, but the likelihood probability estimation is often very difficult to compute due the complex structure of the human body. The computational complexity of this approach is also very high. Among the different rule based technique, Shet et al. [163] apply three distinctive set of rules: detector based, geometry based and explanation based. Ramanan et al. [151] use clustering on the detected body parts and prune the non-moving clusters to have the final body part combination as full human. In this work, we use non-probabilistic rule based approach (Algorithm 2), inspired by [130] which uses simple geometric rules to remove the false positives generated by the body-part detectors. We define the detected body-part component bounding boxes as, Head Bounding Box (R_H), Leg Bounding Box (R_L) and Arm Bounding Box (R_A) obtained from body-part detectors and the full human bounding box (R_F). Figure 3.6 shows the schematic diagram of our geometric constraint based full human detection system. Different geometric constraints on body-parts are shown in the Figure 3.7.

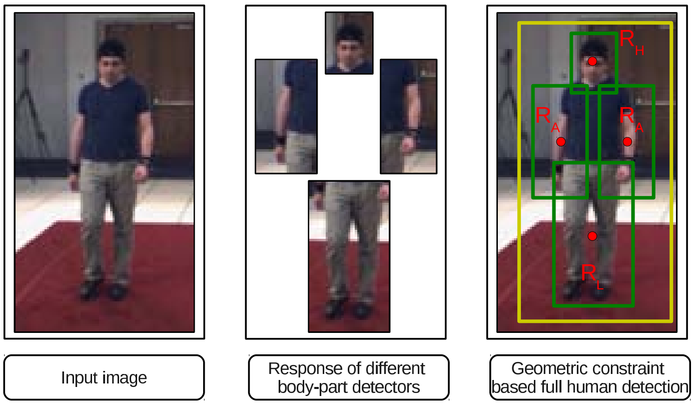

R_H
R_A
R_A
R_L
Input image
Response of different body-part detectors
Geometric constraint based full human detection

Figure 3.6: Schematic diagram of our geometric constraint based full human detection.

Algorithm 2 Geometric constraints on body-part detection

```
Require: Detected body-part bounding boxes: $R_H$, $R_L$ and $R_A$;
    Width and height of full human bounding box: $W_{R_F}$ and $H_{R_F}$.
Ensure: Full human bounding box: $R_F$, if geometric constraints are satisfied;
    NULL, otherwise.
$(X_{C_H}, Y_{C_H}) \leftarrow$ CENTROID($R_H$).
$(X_{C_L}, Y_{C_L}) \leftarrow$ CENTROID($R_L$).
if $(X_{C_H} - \frac{W_{R_H}}{2}) < X_{C_L} < (X_{C_H} + \frac{W_{R_H}}{2})$ then
    if $Y_{C_L} < \frac{H_{R_F}}{2}$ then
        Obtain $R_F$ using $\{(X_{C_H} - \frac{W_{R_F}}{2}), (Y_{C_H} + \frac{H_{R_H}}{2}), W_F, H_F\}$.
        $(X_{C_F}, Y_{C_F}) \leftarrow$ CENTROID($R_F$).
        $(X_{C_A}, Y_{C_A}) \leftarrow$ CENTROID($R_A$).
        if $(X_{C_F} - \frac{W_{R_F}}{2}) < X_{C_A} < (X_{C_F} + \frac{W_{R_F}}{2})$ then
            if $Y_{C_H} > Y_{C_A} > Y_{C_F}$ then
                Return($R_F$).
            end if
        end if
    end if
else
    Return(NULL).
end if
```

Removal of false positives is depicted in Figure 3.8. When the detectors are applied in the image there are usually many overlapping detected windows for a particular body component. We obtain a single bounding box from those overlapping detected windows in the following way. If two bounding boxes share more than 70% overlapping area they are merged to obtain one single box. In this way overlapping detection

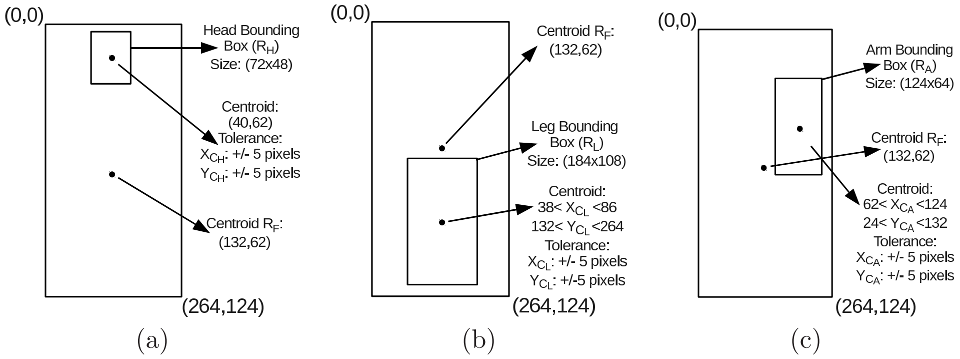


Figure 3.7: Geometric constraints that are placed on the different body-parts. All coordinates are relative to the upper left-hand corner of (264 × 124) full human bounding box, R_F. (a) Illustrates the geometric constraints on the head boundig box, R_H, (b) the leg bounding box, R_L and (c) right arm bounding box R_A.

windows are converted into a single one. After that, the above geometric constraint is applied to remove possible false positives and we obtain an ensemble of body-part detectors resulting in a full human detection.

3.3 Action recognition using HMMs

We choose Hidden Markov Models for modelling $\Pr(a_i|d(a_i), I)$ (Equation 3.3) where $d(a_i)$ is either B_{legs} or B_{arms}. That is, we use $d(a_i)$ to indicate whether action a_i depends mostly on the arms or on the legs.

3.3.1 Definition of action HMMs

An HMM is a collection of finite states connected by transitions. Each state is characterized by two sets of probabilities: a transition probability and either a discrete output probability distribution or a continuous output probability density function. These functions define the conditional probability of emitting each output symbol from a finite alphabet, conditioned on an unknown state. More formally, it is defined by: (1) A set of states S; (2) The transition probability matrix, $T = \{t_{ij}\}$, where t_{ij} is the probability of taking the transition from state i to state j and (3) The output probability matrix R. For a discrete HMM, $R = \{r_j(O_k)\}$, where O_k represents a discrete observation symbol. The initial state distribution is $\pi = \{\pi_i\}$, and the complete parameter set of the HMM can be expressed as:

$$\lambda = (T, R, \pi). \tag{3.6}$$

Here, for each action a_i one discrete HMM is constructed using features from the contributing body parts: B_{legs} or B_{arms}. We obtain the set of hidden states, S, using

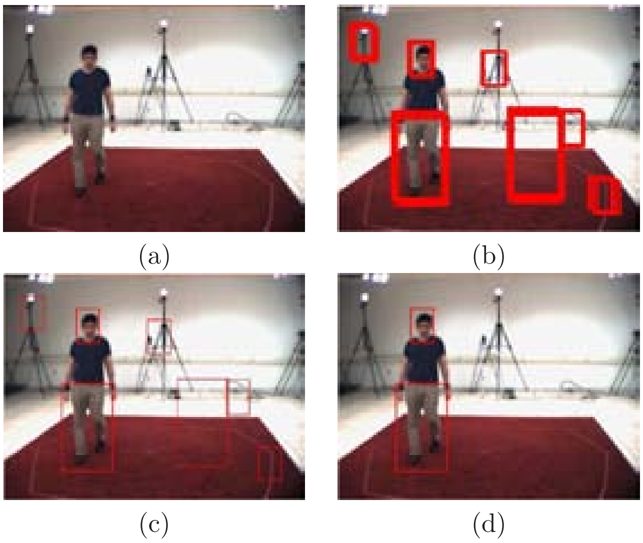


Figure 3.8: Removal of false positives using geometric constraints of different component detectors. (a) original image (b) detection of head and legs with overlapping bounding boxes (c) after getting single detection window for head and leg including false positives (d) detection of head and leg after removal of false positives using geometric constraint.

a Gaussian Mixture Model (GMM) on the features from the detected body-parts. The transition probability matrix, T, is learnt and action classification is done after computing the output probability matrix, R, accordingly.

3.3.2 Construction of action HMMs

For learning a particular action HMM, several sequences of frames are chosen and every such sequence is called a *cycle* of that action. The number of frames that define one *cycle* depends on the training sequences and the action itself. For example, the number of frames in an action cycle varies when it is performed in a circular path. Let there be M frames inside one action cycle and in each of these frames the body parts, B_{legs} or B_{arms}, are detected using component detectors (see Section 3.2). Assuming that in the kth frame the detected bounding box of body-part, B_{legs} or B_{arms}, has a total of n best $6D$ feature vectors from Section 3.2.1 as $\{G_1^k, G_2^k, ..., G_n^k\}$ where each $G_i^k = \left\langle g_1^i, g_2^i, \ldots, g_6^i \right\rangle^k$. We compute the mean over all these G_i^ks to get the features from the kth frame for HMM learning. So, we have $\langle \mu_1, \mu_2, \ldots, \mu_M \rangle$ as the feature set to construct the HMM where each of these μ_ks, $k = 1, 2, \ldots, M$ is,

$$\mu_k = \frac{1}{n} \times \sum_{i=1}^{n} \left\langle g_1^i, g_2^i, \ldots, g_6^i \right\rangle^k . \tag{3.7}$$

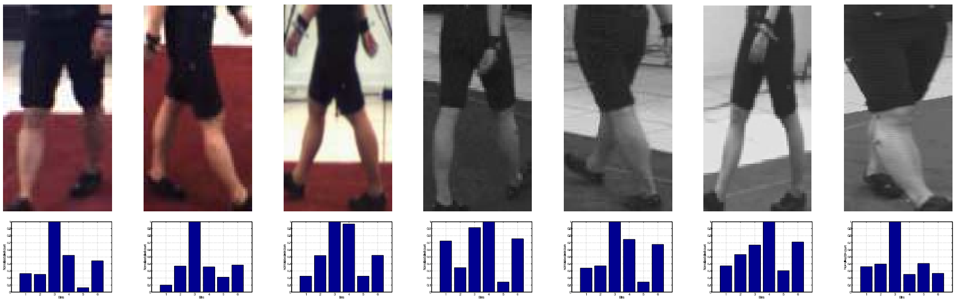


Figure 3.9: Taking the mean of the HOG features from selected feature bins. These figures contain views of the leg body part along with its feature representation computed as the mean of the HOG cells corresponding to detected features. These examples illustrate how the mean results in a discriminative feature representation for distinct leg poses.

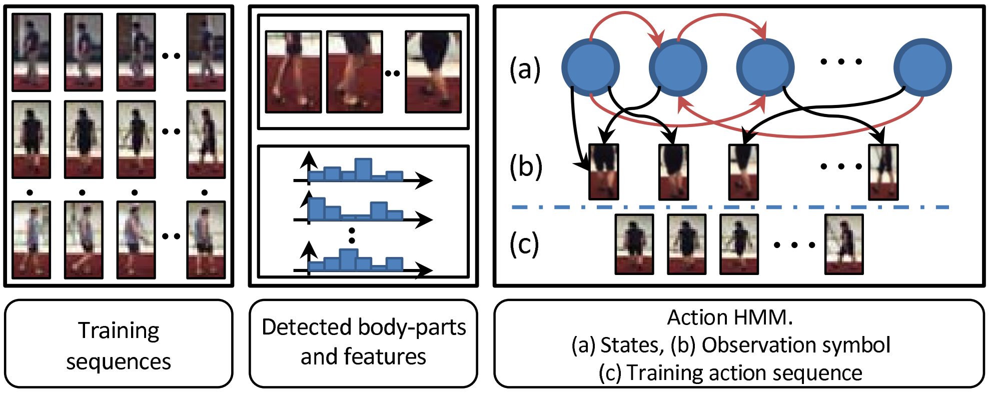

(a)
(b)
(c)
Training sequences
Detected body-parts and features
Action HMM.
(a) States, (b) Observation symbol
(c) Training action sequence

Figure 3.10: Action HMM learning frame work.

In each μ_k of Equation 3.7, the bins with strongest responses provide the general orientation of the body part which in turn signifies one pose or a series of similar poses in an action. Figure 3.9 illustrates how using the mean results in a discriminative representation of body parts. This is important as we want distinct body part poses to end up in distinct clusters during the key-pose alphabet learning phase which follows.

After obtaining features from each training action sequence, we fit a Gaussian Mixture Model (GMM) using Expectation Maximization (EM) to obtain Q *key-poses* which form the basic symbols in the alphabet of the action HMM. Note also that $Q < P$ must hold, where P is the total number of body parts detected from all the training sequences. These *key-poses* are the center of each cluster s_i of $S = \{s_1, s_2, \ldots, s_Q\}$, obtained from the GMM. Now, the prior π (Equation 3.6) of each state, π_{s_i}, is,

$$\pi_{s_i} = \frac{\#s_i}{P}. \tag{3.8}$$

For each training sequence, we compute the features, as in Equation 3.7, from the detected body-part in each frame as the observation symbol O_k. We use the Baum-Welch algorithm [57, 149] to estimate the other two parameters of action HMM, the state transition matrix T and the output probability matrix R (Equation 3.6). Figure 3.10 shows the framework of our action HMM.

For an unknown action sequence, we detect features (Equation 3.7) from the body poses in all frames as the unknown observation sequence, $O_{te} = \{O_{te_1}, O_{te_2}, \ldots, O_{te_V}\}$, where V is the total number detected body parts in all the frames. We use the Forward algorithm [57, 149] to compute the joint-probability $\Pr(O_{te}|\lambda_{a_i})$, where λ_{a_i} is the HMM of the action a_i, as the probability, $\Pr(a_i|d(a_i), I)$ of Equation 3.3. The action class that gives the maximum of these probability values (Equation 3.4), is the class label of a unknown action sequence.

3.4 Experiments

We use four publicly available datasets for our experiments on body-part detection and action recognition. HumanEva dataset[1], KTH dataset[2] and HERMES indoor dataset[3], Weizmann dataset[4]. Please refer to the Chapter A for the detailed description of these datasets.

3.4.1 Training of the body part detectors

In our method for head, leg and arm detection the bounding box sizes are fixed to (72×48), (184×108) and (124×64) pixels respectively. A (264×124) pixel bounding box is applied for full human. Since the test image is zoomed to various sizes, and in each zoomed image the components of those sizes are searched, the fixed component sizes do not affect scale invariance of human detection. To train each component detector, $10,000$ true positives and $20,000$ false positives are selected from the HumanEva dataset are used. The sizes of different body-part component bounding boxes are determined from the the statistics of height and width of each component from all the sequences of the HumanEva dataset. A tolerance of 10 pixels is also used on the height and width of each bounding box.

3.4.2 Performance evaluation of body-part detection

We performed extensive experiments and quantitative evaluation of the proposed approach to body part detection. We validate our part detector using HumanEva dataset. For a particular body-part detector we typically obtain different detection

[1] http://vision.cs.brown.edu/humaneva/

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

[3] http://iselab.cvc.uab.es/indoor-database

[4] http://www.wisdom.weizmann.ac.il/∼vision/SpaceTimeActions.html

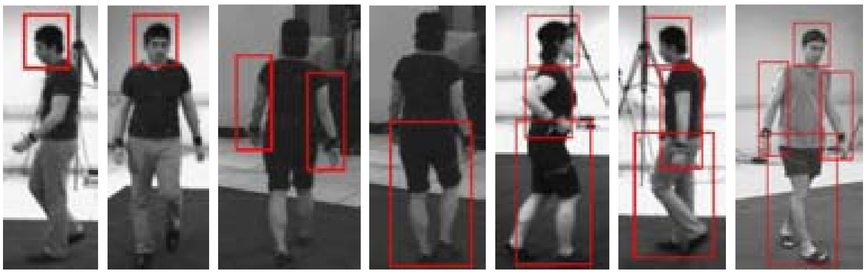


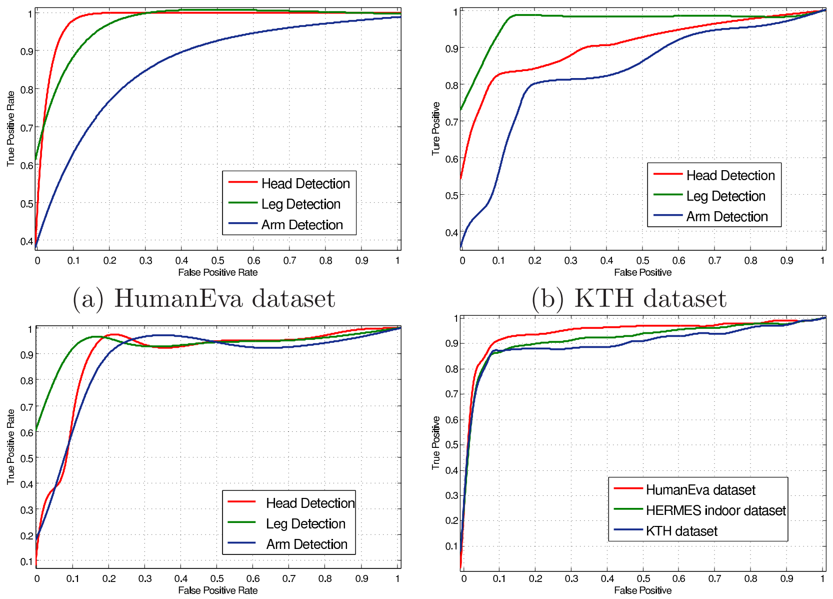
Figure 3.11: Results of head, arm and leg detection on the HumanEva dataset. Here detection of profile head and leg are shown. Arm detections are shown where both arms are visible and one is occluded.

scores from the sub-classifiers that the detector is composed of. We choose the detection result based on the best detection score among themselves. Figure 3.11 shows the results of the component detectors on the HumanEva dataset. These detections illustrate the view invariance of our detectors since in the HumanEva dataset the agents perform actions in a circular path. To test the performance of body part detectors, the KTH Dataset [159] and HERMES indoor sequence dataset [62] are used.

Receiver Operating Characteristic (ROC) curves are shown in Figure 3.12 for three component detectors, head, legs and arms, on different datasets. Figure 3.12 also contains the ROC curve for detection of full humans. These ROC curves are generated considering the overall performance of all the sub-classifiers of each group of the three classifiers. ROC analysis reveals that head detection and leg detection are quite accurate. Although there are false positives, the geometric constraint eliminates them. For the arms, however, the detection rate is not very high since arm pose varies dramatically while performing actions. ROC curves for the full human are computed only for those cases where all three body-parts are detected (possibly with false positives).

The primary objective of our method is human action recognition and not human detection. Geometric constraints are only needed to prune the false positives of the body-part detectors, hence it is more important to show the accuracy of the proposed geometric constraints based method in those cases where all the body-parts are correctly detected. In most cases, in fact, it is preferable to be conservative with detection of full-humans (low false positive rate). Action HMMs are robust to missed frames, but false positive detections could potentially wreak havoc with action recognition by corrupting the stream of key-pose observations. Figure 3.12 illustrates that out technique for pruning false positives using geometric constraints is an effective way of obtaining high true positive rates, while eliminating false positives.

In the KTH dataset, there are several image sequences where it is impossible to detect arms due to different clothes than the other two datasets. In such cases we are able to detect head and legs, and when they are in proper geometrical arrangement, the full human bounding box is constructed. There are some sequences where only



(a) HumanEva dataset (b) KTH dataset

(c) HERMES indoor sequences (d) Full human detection

Figure 3.12: ROC curves for different body part detectors and full human detection on various datasets. The false alarm rate is the number of false positive detections per window inspected. These figures are best viewed in colour.

legs are detected due to low resolution. Figure 3.13 shows some examples of human detection on the KTH dataset. In those images the detected bounding boxes are found at different scales and they are drawn after rescaling them to 1:1 scale. For low resolution image sequences much information has been lost due to scaling and the Sobel mask hardly finds important edges from particular body parts. Our system works well in high resolution datasets like, the HumanEva (resolution 644 × 484) and the HERMES indoor sequences (resolution 1392 × 1040). It gives an average 97% recognition rate for walking, jogging and boxing actions. However, on low resolution datasets like the KTH (resolution 160 × 120) we achieve lower performance on the actions like jogging and hand clapping. In the KTH dataset there are many cases where the human is visible at the original resolution but when it is zoomed to 640×480 to detect the body-parts, the objects are blurred and detection suffers.

3.4.3 Action recognition

For action recognition we use the HumanEva, KTH and Weizmann dataset for evaluation. For each dataset we use a random sample of 50% of the video sequences as training and the rest as test. All recognition rates are computed by averaging over 50 random draws of these training and testing sets for the KTH and HumanEvan experiments, and 25 random training/testing draws on the Weizmann dataset.

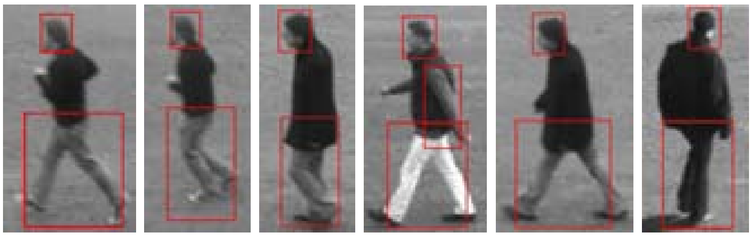


Figure 3.13: Performance of body part detectors on the KTH dataset. Detection of head, arms and leg shown for profile poses.

Table 3.1: Comparison of mean classification accuracy on the HumanEva dataset. Classification accuracy of our method is based averaging over 50 random splits of training and test sets. Note that, the accuracy of [135] is the mean accuracy rate of 6 different models of Conditional Marcov Random Fields and the accuracy rate of [214] is the mean accuracy of 7 different view-points.

Our approach	**90.16**%
Ning et al. [135]	89.87%
Yoon et al. [214]	88.71%

HumanEva dataset We obtain **90.16**% recognition rate for this dataset. Table 3.1 shows a comparison of our recognition rate with other state of the art methods. The recognition accuracy of [135] is the mean accuracy of 6 different models of Conditional Random Fields, using four actions, boxing, walking, jogging and gesture. The best obtained accuracy is 95% out of 6 different models. The accuracy rate of [214] is the average accuracy of 7 viewpoints of HumanEva and only three actions, boxing, walking and jogging, are used. The recognition rate obtained in our approach is computed using all five actions, boxing, walking, jogging, gesture and throw catch of HumanEva and we get higher accuracy rates.

Table 3.2 shows the confusion matrix. We have major confusions in boxing and throw-catching actions. These confusions are mostly due to the difficulty in arm detection in these actions. We use all the 7 view-point videos for action recognition. This wide range of view-point usually causes partial occlusion of the body-parts. Our sub-classifiers are able to handle this problem of partial occlusion and the high recognition rate shows that our approach can perform action recognition in different view-points.

KTH dataset In this dataset we get **83.50**% recognition rate. Table 3.3 shows a comparison of our recognition rate with other methods. The average action recognition rate obtained from our method is promising but fails to surpass some of the state-of-the-art methods, [152, 95, 80, 104, 211, 109, 113].

Out of these methods, Lin et al. [109] use a prototype tree based search approach to find the action where different motion based features are used to build the action

Table 3.2: Confusion matrix of the HumanEva dataset. Confusions in % are based on averaging over 50 random splits of training and test sets.

	Walking	Jogging	Boxing	Gesture	Throw-catch
Walk	**97.3**	2.7	0.0	0.0	0.0
Jogging	8.8	**91.2**	0.0	0.0	0.0
Boxing	0.0	0.0	**83.7**	14.0	2.3
Gesture	0.0	0.0	0.0	**96.2**	3.8
Throw-catch	0.0	0.0	0.0	17.6	**82.4**

prototype tree and rest of the approaches are based on spatio-temporal interest points (STIPs). Among these STIP-based methods, [152, 95, 80, 104, 109, 113] most use bag-of-words model over features extracted from detected STIPs, while Yao et al.[211] perform STIP tracking using a particle filter. A distinguishing feature of these approaches is that they all use motion and motion models in some way. Our approach does not explicitly use any motion information, but rather learns the deformation of body-parts during the action through action HMMs. It is worth noting that without using any direct motion modelling or motion based features, we achieve 83.50% recognition rate on the KTH dataset. Table 3.4 shows the confusion matrix of our approach on the KTH dataset. In the confusion matrix we can see that our approach can distinguish the leg and arm based actions. Among the actions, hand clapping have the poor recognition rate. In this case, we confuse the other two arm based actions: boxing (13.5%) and hand waving (19.8%). Most of the errors on the KTH dataset are due failures in part detection caused by the low resolution of the KTH dataset.

Table 3.3: Comparison of mean classification accuracy on the KTH dataset. Classification accuracy of our method is based averaging over 50 random splits of training and test sets. For other methods it is the accuracy number reported in the referenced paper.

Ke et al. [91]	62.90%
Wong et al. [203]	71.16%
Schuldt et al. [159]	71.72%
Dollár et al. [44]	81.17%
Niebles et al. [134]	81.50%
Our Approach	**83.50**%
Reddy et al. [152]	90.30%
Kläser et al. [95]	91.40%
Jhuang et al. [80]	91.70%
Laptev et al. [104]	91.80%
Yao et al. [211]	93.00%
Lin et al. [109]	93.43%
Liu et al. [113]	94.16%

Table 3.4: Confusion matrix of the KTH dataset. Confusions in % are based on averaging over 50 random splits of training and test sets.

	Walking	Jogging	Running	Boxing	Waving	Clapping
Walking	**95.0**	4.0	1.0	0.0	0.0	0.0
Jogging	7.0	**89.0**	4.0	0.0	0.0	0.0
Running	0.0	23.1	**76.9**	0.0	0.0	0.0
Boxing	0.0	0.0	0.0	**100.0**	0.0	0.0
Waving	0.0	0.0	0.0	20.0	**73.4**	6.6
Clapping	0.0	0.0	0.0	13.5	19.8	**66.7**

Weizmann dataset Here we obtain a recognition rate of **97.85%**. We exclude the bend action for the test, since our full human detection algorithm (see Section 3.2.3) can not recognize humans in bending posture since these deformations are not included in the HumanEva set used for training part detectors. The confusion matrix in Table 3.5 shows that 2.5% of the action wave2 is detected as the action jack. This is due to the similarity in arm movement in these two actions. We provide a comparison table (Table 3.6) putting the recent state-of-the-art methods. Our accuracy rate is near the state-of-the-art, although we can see that on the Weizmann dataset reported accuracy rates have reached 100%. Again the methods [80, 63, 109], having higher recognition rate than us, use template motion as a primary feature for action modelling.

Table 3.5: Confusion matrix of the Weizmann dataset. Confusions in % are based on averaging over 25 random splits of training and test sets.

	Jack	Jump	PJump	Run	Side	Skip	Walk	Wave1	Wave2
Jack	**100.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Jump	0.0	**100.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PJump	0.0	5.0	**95.0**	0.0	0.0	0.0	0.0	0.0	0.0
Run	0.0	0.0	0.0	**98.3**	0.0	0.0	1.7	0.0	0.0
Slide	0.0	0.0	0.0	0.0	**100.0**	0.0	0.0	0.0	0.0
Skip	0.0	3.4	0.0	0.0	0.0	**96.6**	0.0	0.0	0.0
Walk	0.0	1.7	0.0	5.0	0.0	0.0	**93.3**	0.0	0.0
Wave1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**100.0**	0.0
Wave2	2.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**97.5**

3.4.4 Computational time of the proposed method

The proposed action recognition system can be divided into two basic parts: 1) detection of body-parts and combining them using geometric constraint to a full human; and 2) joint probability computation using action HMMs. The processing time of our human detection system is 3 frames/second on a standard dual core Desktop PC (Intel(R) Core(TM)2 CPU 6400@2.13GHz 2GB RAM), having frame resolution

Table 3.6: Comparison of mean classification accuracy on the Weizmann dataset. Classification accuracy of our method is based averaging over 25 random splits of training and test sets. For other methods it is the accuracy number reported in the referenced paper.

Kläser et al. [95]	84.30%
Yao et al. [211]	92.20%
Ali et al. [6]	92.60%
Our approach	**97.85%**
Jhuang et al. [80]	98.80%
Gorelick et al. [63]	99.64%
Lin et al. [109]	100.00%

(120×160). The action HMMs process 5 sequences per minute, where each sequence contains about 500 frames. The code for body-part detectors and HMMs are implemented using C++ and Matlab, respectively. Neither module has been extensively optimized for speed.

Not many state-of-the-art methods discuss the computational time of their proposed methods. Wang et al. [190] has recently reported the computational time of different spatio-temporal feature extraction methods. They show that the fastest method is Hessian+ESURF, which computes 44 features at 4.6 frames/second. In our case only feature extraction works quite fast. It computes nearly 1000 6-bin HOG features per frame at a speed of 4 frames/second which is comparable to the time of Hessian+ESURF detector. The computational time of our action recognition system is comparable to that of Jhuang et al [80] which processes a single video of 50 frames in 2 minutes. This comparison shows that although our system does not work in real-time (5 videos/minute) due to the usage of the HMMs but still the computational time is on par with other state-of-the-art methods.

3.5 Discussion and Conclusion

This work presents a novel approach for recognizing actions based on a probabilistic optimization model of body limbs. It also uses view-point invariant human detection and example-based body-part detectors to model the characteristic poses of humans as they execute various actions.

Stochastic changes of body components are modelled using HMMs. This method is also able to distinguish very similar actions like walking, jogging and running (considering features from legs); boxing, hand waving and hand clapping (considering features from hands). The leg movements, sometimes, are very similar in the actions like jogging and running. Also, in some cases there are problems of resolution and contrast which makes it difficult to distinguish those actions. Actions, involving hand motion also suffer from similar problems, some part of hand waving action look similar to hand clapping, which in turn causes ambiguity.

We observe that in the KTH dataset there are major confusions occurred between clapping and waving action (Table 3.4) and for the HumanEva dataset (Table 3.2) the confusions occur in boxing and throw-catching actions. This is because of the arm detectors are difficult to design for every possible degree of freedom specially in those two actions there are a variety of arm pose changes. On the other hand, arm detectors perform well for boxing since in this case the pose changes do not vary a lot. The advantage of our approach is two fold; first, the body part detection is robust except for resolution limited images. Second, the HMM-based action model is capable of recognizing actions even when the body part detectors fail on some frames of an action sequence. Other important aspect of the body-part based action HMMs is that it never confuses between the leg and arm based action, which is often be the case for other action recognition approaches like, the spatio-temporal interest point based approach.

Proposed approach does not beat the best state-of-the-art datasets on the KTH or Weizmann datasets. However we would like to point out that our approach has the advantage that our features for action classification are learned only once on an independent dataset. That is, we only learn part models on the HumanEva and our experiments show that this generalizes to other datasets with different actions and different illumination conditions. It is also worth noting that all of our accuracy values we report are based on multiple random splits of training and testing (50 random draws for KTH and 25 for Weizmann). Most published numbers of these datasets are either based on a single split, or the authors do not specify how their splits are done.

In the current system, the automatic learning of major contributing body-part detection is absent and this is an interesting direction for future research. In the training process we learn HMMs using features from leg or arms depending on the action being learned. Automatic learning can be introduced by applying tracking on the detected body parts to identify the moving parts contributing most to a specific action. The performance can be improved in human detection by adding more training samples, by including more pose deformations in training and introducing more angular views. There is no good dataset for different body-part components, so building a component dataset is an important task. For action recognition, higher order HMM can be applied. Information of other body parts which have minor contribution in the action, like arms for walking, can be included in order to minimize the misclassification rate.

Chapter 4

Limb dynamics based key pose for action recognition

"One very important aspect of motivation is the willingness to stop and to look at things that no one else has bothered to look at. This simple process of focusing on things that are normally taken for granted is a powerful source of creativity..."

by Edward de Bono

Exploring visual information is an important aspect in action recognition. Based on the visual information, an in depth understanding of the limb dynamics can be achieved for a robust action representation. Key pose identification is another aspect for action recognition. This chapter explains the idea of key pose detection using k-medoids clustering techniques on the limb dynamics. The limb dynamics are computed using circular statistics on Histogram of oriented gradient (HOG) features. Different actions are represented by the histogram of Bag-of-key poses. Finally a Support Vector Machine is learned using the histograms and actions are classified.

Recognizing human action using minimum information is an interesting area to explore. Recent work of Schindler et al. [157] study the problem of finding minimum number of frames to recognize human action. By using both visual and dynamic cues they conclude that "basic action can be recognized well even with very short snippets of $1 - 7$ frames (at frame rate 25 Hz). In [197], Weinland et al. works on human action recognition with pure visual cues and they used a whole sequence to get a classification result. Their conclusion is that it is possible to recognize human actions with pure visual cues except for some special cases like and action and its reversal. Other works, like in [192, 11], also confirm the similar observation on human action recognition. Wang et al. [192] use a compact representation of the actions by exploring the directional data of the Histogram of Oriented Gradient (HOG) [40] features extracted from human silhouette. Baysal et al. [11] use a key pose identification technique from a sequence of images extracted from and action video. Using nearest neighbour based approach on these key poses action recognition is done.

Motivated by the above ideas, in this chapter we use limb dynamics to find the key-poses of an action. Continuing with idea of the previous chapter (Chapter 3) a human detector is applied to identify the full human in the scene. A synthetic average exact filter (ASEF)[17, 18] based real time human detector is used for this purpose. Limb dynamics are then computed by using circular dynamics of HOG features described in [192]. We use the idea of Bag-of-Words model [169, 134, 38] and represent action using Bag-of-Key poses to obtain a key-pose histogram. Using Support Vector Model (SVM) action model is trained and classified. Figure 4.1 shows a schematic digram of the proposed framework.

The outline of this chapter is as follows. ASEF based human detection is described in Section 4.1. Section 4.2 explains the limb dynamics feature computation. Key-pose estimation and Bag-of-Key pose representation is elaborated in Section 4.3. Experimental results are shown in Section 4.4. Finally, the chapter concludes with a brief discussion in Section 4.5.

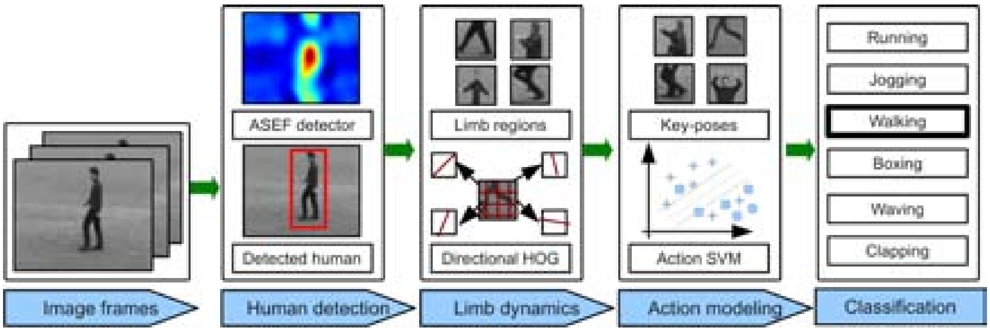


Figure 4.1: Schematic digram showing weak model based human action recognition. An ASEF based real time human detector is applied to identify the upper and lower limb regions. Directional HOG features are computed to model the limb dynamics. A *K-medoids* based clustering algorithm is applies on the limb dynamics features to exptract *key-poses*. Actions are represented as a Bag-of-key-posses and SVM is applied for action classification.

4.1 ASEF for Human Detection

Average synthetic exact filter (ASEF) was first proposed by Bolme et al. [17] for eye detector and is used as a real time human detector in [18]. We use the similar framework for the identifying human in a video. A bank of ASEF learn the mapping from a source image to a target image. More formally, given an image $f \in \mathbb{R}^{P\times Q}$, ASEF map it to a new image $g \in \mathbb{R}^{P \times Q}$ and this mapping is parameterized by a filter $h \in \mathbb{R}^{P \times Q}$ and the transformation can be expressed as a convolution:

$$g = f \otimes h \tag{4.1}$$

To train a standard human detector, each detection windows is labeled as either being a positive example if a human is present or a negative example if it corresponds to background. In contrast, the ASEF is trained on complex scenes that contain both positive and negative samples. The entire image is labeled with peaks with values of 1.0 where a person is present and values of 0.0 for background. The ASEF process learns a mapping from the training images to the labeled outputs.

For each training image f_i, a synthetic output g_i is generated which contains a peak for each person in the image. The peaks in g_i take the shape of two dimensional Gaussians:

$$g_i(x, y) = \sum_p exp^{-\frac{(x-x_p)^2+(y-y_p)^2}{\sigma^2}} \tag{4.2}$$

where (x_p, y_p) is the location of person p in the training image, and σ controls the radius of the peak.

Next, for each training image, an exact filter h_i is computed which is exactly maps the image f_i to g_i. This computation is efficient in Fourier domain.

$$g_i = f_i \otimes h_i \tag{4.3}$$

In frequency domain convolution is element-wise multiplication. Therefore, the filtering operation can be transformed as:

$$G_i = F_i \odot H_i \tag{4.4}$$

in Fourier domain, where G_i, F_i and H_i are the Fourier transforms of their lower case counterparts and $\odot$ indicates an element-wise multiplication. The exact filter [17], H_i can be computed by solving Equation 4.4:

$$H_i = \frac{G_i}{F_i} \tag{4.5}$$

where the division is also performed element-wise.

The resulting filter could be considered as a weak classifier that performs perfectly on a single training image. It does not, however, generalize well to the larger dataset. As seen in Figure the single exact filter looks more like noise than a template that will respond to a person's outline. To obtain a more general classifier, exact filters are computed for every training image and then averaged [21]. Aggregating a collection of simple filters converges on a filter that minimizes the variance error. Therefore, the final ASEF filter is computed as:

$$h = \frac{1}{N} \sum_{i=1}^{N} h_i = \frac{1}{N} \mathcal{F}^{-1} \left(\sum_i H_i \right) \tag{4.6}$$

where N is the number of training images. ASEF for human detection is quite fast to and easy to compute: it does not over-fit the training data, it only requires a single pass for each image.

As mentioned in [18], ASEF requires a large number of training images, as a negative aspect. But in our case, we use $10K$ image frame from KTH dataset to design the human detector (See Section 4.4). Other important issue regarding the filter design is, in Equation 4.5, frequencies in the training f_i that contain very low energy are weighted heavily in corresponding exact filter. This makes the final exact filter unstable, in extreme case where the frequency is zero cause a divided by zero error. To correct this problem, the exact filters are constructed using the largest frequencies in F_i that contain 95% of the total energy. Removing the small frequencies appears to remove a greater percentage of "noise" in the exact filter. Figure 4.2 shows the ASEF human detector. Figure 4.2.a depicts the training images and corresponding Gaussian model as a synthetic output. Actual ASEF for human detection is shown in Figure 4.2.b, as mentioned above, we use 95% of the total energy to construct the filter. ASEF performs real-time with robust detection performance. But, it can not adapt to the scale variations. To cope with this, we propose n different ASEF human detectors with varying scale. The number n can be chosen according to the dataset and the scale range of the human appears in the scene. These filters are applied simultaneously in the image frame and the detected region is verified using a verification human detector similar to [40], which is designed by using HOG features and SVM. This extension helps to reduce the entire sliding window search technique

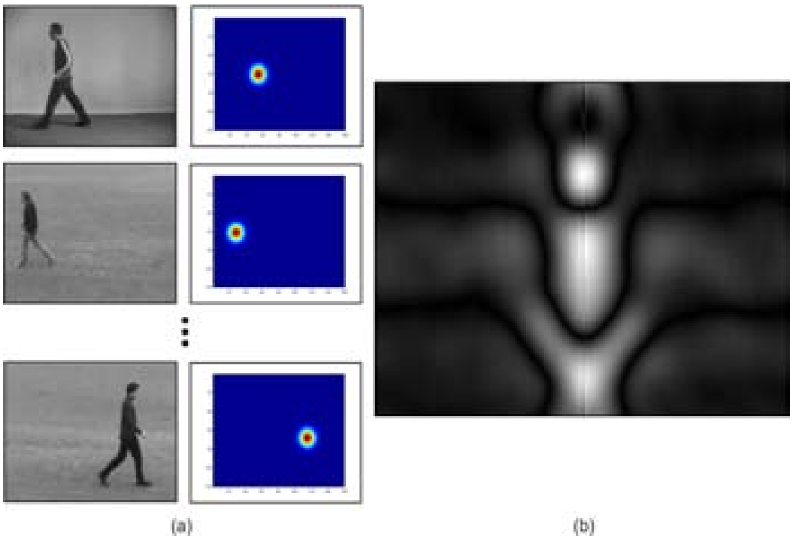


Figure 4.2: Design of an ASEF human detector. (a) The training images and corresponding synthetic Gaussian output centered in the human. (b) Actual ASEF human filter, where 95% of the total energy is used to avoid the greater percentage of "noise".

originally used by Dalal et al. [40]. Figure 4.3 describe this process. From the initial hypothesis, red bounding box of Figure 4.3.c which is derived from the ASEF filter output (Figure 4.3.b) we obtain the possible area to search for human center (blue rectangle). Through out this area possible human bounding box is searched (green rectangle) using the human detector similar to [40]. Finally the human is obtained as in Figure 4.3.d.

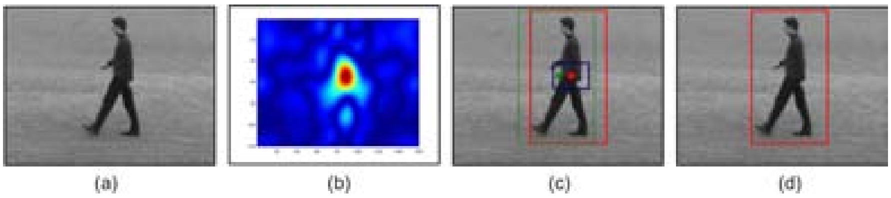


Figure 4.3: Overview of the scale adaptive ASEF human detector. (a) The image frame with human (b) the ASEF output, (c) initial hypothesis of human as red bounding box. Blue rectangle is the area where the center of the possible human bounding box may occur. This is verified by a human detector similar to [40] and green bounding is showing this process. (d) Finally, the detected human.

4.2 Limb dynamics extraction

The detected human bounding box is divided into upper and lower parts to obtain the limb dynamics features. In this work we use directional statistics of HOG features similar to [192]. In this way, we are capturing different pose variations during actions.

4.2.1 Directional statistics

Directional statistics is a discipliner of the statistics that deals with directions (unit vectors in $\mathbb{R}^n$), axes (lines through the origin in $\mathbb{R}^n$ or rotation in $\mathbb{R}^n$). More generally, directional statistics deals with observations on compact Riemannian manifolds[1].

The feature of directional data does not have specific start or end point. This leads to incorrect mean and standard deviation using regular statistics. For example, averaging 1° and 359° to be 180° with regular statistics. With directional statistics, rotation invariant statistics can be obtained.

Two basic descriptive statistics in directional statistics are mean direction (MD) and circular standard deviation (CSTD). The mean direction $\bar{\theta}$ and the circular standard deviation σ_0 are calculated using Equation 4.10 and Equation 4.11:

$$\bar{C} = \frac{1}{n}\sum_{j=1}^{n} cos\theta_j \tag{4.7}$$

[1] http://en.wikipedia.org/wiki/Directional_statistics

Window Height	64	Window Width	48
Block Height	16	Block Width	16
Cell Height	8	Cell Width	8
Bin size	9	Direction range	$0 - \pi$

Table 4.1: Parameter settings of HOG computation.

$$\bar{S} = \frac{1}{n}\sum_{j=1}^{n} sin\theta_j \tag{4.8}$$

$$\bar{\theta} = arctan\left(\frac{\bar{S}}{\bar{C}}\right) \tag{4.9}$$

$$\bar{R} = (\bar{C}^2 + \bar{S}^2)^{\frac{1}{2}} \tag{4.10}$$

$$\sigma_0 = -2log(\bar{R})^{\frac{1}{2}} \tag{4.11}$$

Direct application of directional statistics on HOG is not straight forward due to following:

1. Directions used in HOG is usually in the range of 0 and π.
2. HOG is a binned representation.

As discussed in [124], for the first issue the unsigned directions can be multiplied by 2 before using directional statistics and divide resultant direction by 2 after. For the second issue, by using bin centers as directions and bin value as weight of each direction, approximation of HOG binning can be obtained with an ignorable difference as long as bin width is smaller than 45^o.

4.2.2 Directional HOG feature and similarity measurement

To extract the direction statistics of HOG features, first we compute HOG from a limb region obtained by applying the ASEF human detector (see Section 4.1) and dividing the detected bounding box into upper and lower parts. The HOG implementation is similar to [40] using the parameters given in Table 4.1.

As described in the Figure 4.4, HOG features are always having overlapping blocks. Thus, from for each HOG cell *four* histograms are obtained. Histograms for non-overlapping parts of boundary cells are set to zero. We compute MD and CSTD of each histogram in HOG to obtain the limb dynamic feature F:

$$F = \begin{bmatrix} F_{11} & F_{12} & \dots & F_{1N} \\ F_{21} & F_{22} & \dots & F_{2N} \\ \dots & \dots & \dots & \dots \\ F_{M1} & F_{M2} & \dots & F_{MN} \end{bmatrix} \tag{4.12}$$

$$F_{ij} = (\bar{\theta}_{ij}, \sigma_{0_{ij}}) \tag{4.13}$$

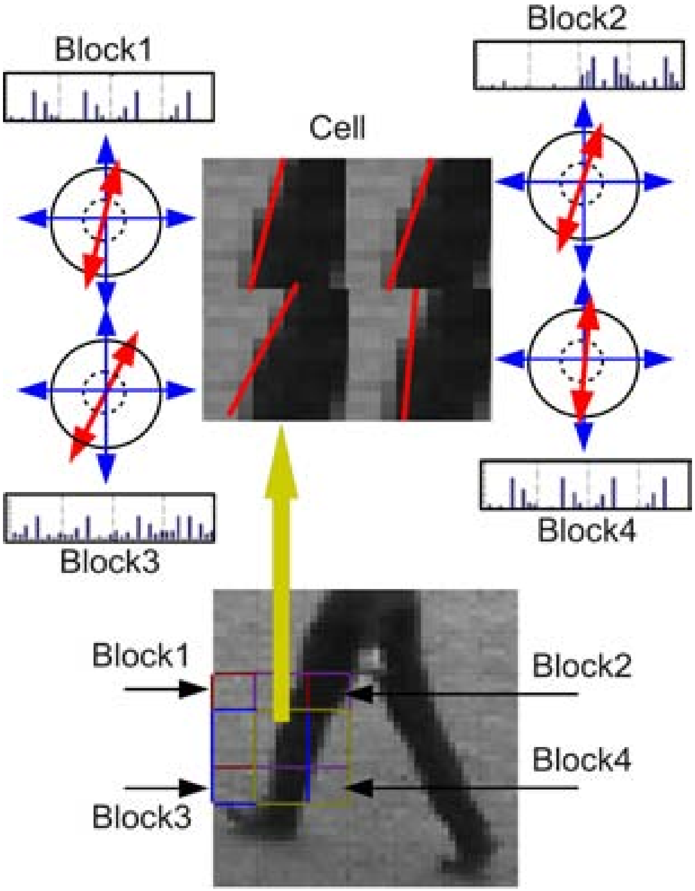


Figure 4.4: Directional HOG feature extraction as limb dynamics feature. 4 HOG blocks are shown using colour boxes. From each block HOGs directional features are computed to obtain the *cell* HOG feature.

Here ij refers to the ij^{th} sub-cell. N is the number of rows of sub-cells and M is the number of columns of the sub-cells. $\bar{\theta}_{ij}$ is MD of ij^{th} sub-cell and $\sigma_{0_{ij}}$ is the CSTD of ij^{th} sub-cell.

To compute the similarity measure of directional HOG, we use similar technique described in [192]. Direction distance of two limb dynamics features are computed by using directional similarity:

$$dist(F_m, F_n) = \frac{\sum_i^N \sum_j^M [(\pi - abs(\pi - abs(\bar{\theta}_{m_{ij}} - \bar{\theta}_{n_{ij}}))) + \frac{(\sigma_{0m_{ij}} > 0)}{2} + \frac{(\sigma_{0n_{ij}} > 0}{2})]}{(N \times M \times \pi)} \tag{4.14}$$

$$sim(F_m, F_n) = exp(-dist(F_m, F_n)^2) \tag{4.15}$$

where F_m and F_n are two limb dynamics features.

4.3 Action Recognition

Next step towards action recognition is to identify *key-frames* using the limb dynamics features from specific actions. Intuitively, to find key-poses, it is reasonable to group the frames, which show common pose appearances. We use k-medoids clustering algorithm to find the group the common poses since the clustering medoids tend to represent common poses in each action.

To identify the true distinguishing key-pose following algorithm 3 is applied. From an action video, limb regions are identified and limb dynamics features are extracted. A k-medoids clustering technique is applied on the limb dynamics feature to have the initial key-poses. To rank the key-poses, again a similarity measurement is computed with all the action image frames.

Algorithm 3 Key-pose ranking

```
Require: Action image frames.
Ensure: Set of key-poses.
$F_{limb} = .$
for Every action training image frames do
    Apply $ASEF_{human}$ on image frame.
    Divide the detected bounding box to obtain limbs.
    $F_{limb} = F_{limb} \cup Direction\_HOG(limbregion)$.
end for
for Each frame $f$ of the training action class do
    Find the most similar key-pose $c_i$ using the Equation 4.15.
    if $label(f) == label(c_i)$ then
        $Score_{c_i} = Score_{c_i} + 1$.
    else
        $Score_{c_i} = Score_{c_i} - 1$.
    end if
end for
Sort the $Score$ to obtain the set of distinguishing key-pose.
```

4.3.1 Bag-of-key poses for action representation

The obtained key-poses are used to represent actions. Let C be the set of action key-poses $C = C_1, \ldots, C_P$, where P is the total number of action classes. The i^{th} training sample video of j^{th} class is represented as a histogram of size $k \times P$, where k is the number of key-poses in each key-pose set C_j. This histogram is computed by the similarity measure between key-poses and the action video frames. The procedure is explained in the Algorithm 4.

Algorithm 4 Histogram of Key-poses

```
Require: Set of key-poses $C = C_1, \ldots, C_P$ having $k$ key-poses in each action class, input video ($v$) and class label.
Ensure: Normalized histogram of key-poses pf size $k \times P$.
Allocate key-pose histogram $hist$ of size $k \times P$.
for Each frame $f$ of the input video $v$ do
    for Each key-pose set $C_i \in C$ do
        Find the most similar key-pose $c_{ij}$ using the Equation 4.15 and the similarity value, $simVal$.
        $hist_v((i-1) \times P + j) = hist_v((i-1) \times P + j) + simVal$
    end for
end for
Normalize the histogram $hist_v$.
```

4.3.2 action SVM

To do the action classification, action class specific Support Vector Machine (SVM) [31, 185] is designed using the histogram of key-poses from each training action class. For i^{th} action class, $SVM_i(K, hist_i^C)$ is designed, where K is the SVM kernel and $hist_i^C$ is the histogram of key-poses computed from the key-pose set C. For a test action A_{Test}, we detect its class,

$$i^*_{A_{Test}} = argmax_j SVM_i(K, hist^C_{A_{Test}}) \tag{4.16}$$

4.4 Experimental Result

We use KTH, Weizmann and CVC action dataset for testing our algorithm. Please refer to the Chapter A for the dataset details.

4.4.1 Human detection

As described in Section 4.1, we have apply a real-time human detector using ASEF. This filter bank usually requires a number of training images to have averaging effect on the background energy components. To design the filter bank, from the training set of the action videos $\sim 9K$ human images are identified for KTH, Weizmann and CVC action dataset. Verification SVM is designed using the same human images. To design the synthetic output of the human for ASEF, we use a standard frame differencing technique and the motion part to obtain a rough estimate of the human location. We use 5 different human resolutions to cope with the scale adaptation property of ASEF.

Human detection results are shown in Figure 4.5. We present some detection from KTH, Weizmann and CVC action datasets. Note that for CVC action data (4^{th} and

5^{th} images of Figure 4.5) contain multiple-hypothesis for human bounding box. With the help of the proposed verification SVM (See Section 4.1) we are able to detect the human. On an average, we obtain 98% detection performance in all these datasets.

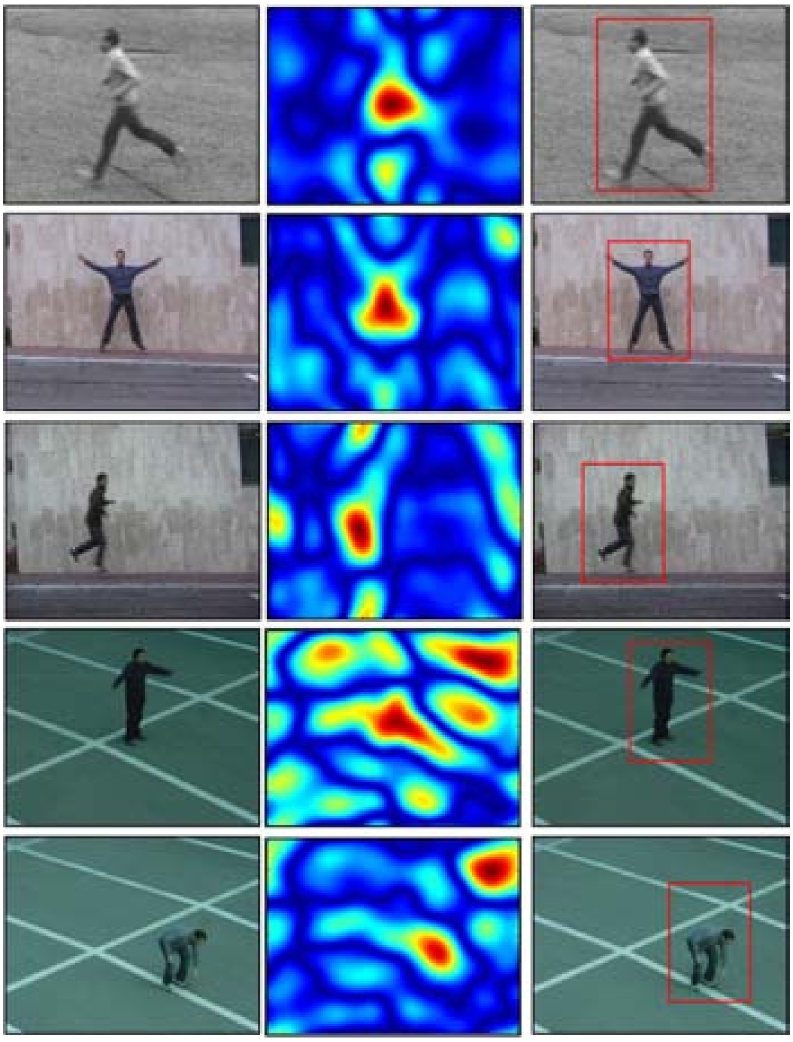
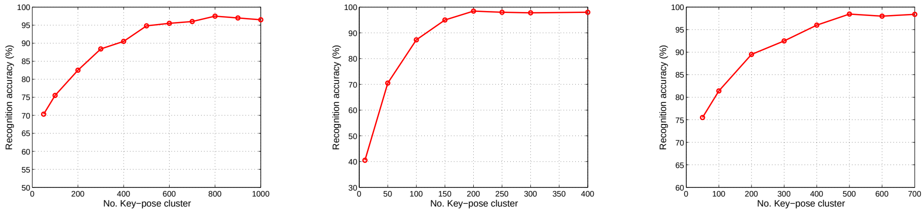


Figure 4.5: Detection result of the ASEF human detector applied in KTH, Weizmann and CVC action datasets. Note that for CVC action dataset (last two images) we obtain multiple hypothesis for initial human from the ASEF filter. But using the verification SVM, the final human identification is obtain.

4.4.2 Action classification

For action classification, a K-medoids clustering algorithm is applied to the limb dynamics features explained in Section 4.2. We performed a test by varying K values and observed the accuracy. Figure 4.6 shows the plot of recognition accuracy as a function of the key-poses. For KTH, 800 key-pose gives the best results. 200 key-poses are the optimum number for Weizmann dataset and for CVC action the value is 500.



(a) KTH dataset (b) Weizmann dataset (c) CVC action dataset

Figure 4.6: Plots showing the recognition accuracy (%) as a function of number of key-poses for (a) KTH, (b) Weizmann and (c) CVC action datasets.

4.4.3 Recognition on KTH:

We apply leave-one-out cross validation (LOOCV) test for KTH dataset. For this we use original dataset division as proposed in [159]. In each LOOCV set up, from the training actions we train the ASEF human detector and extract limb dynamics features both from upper and lower part of the detected human bounding box. Using K-medoids method we extract key-poses from both the limb regions. Valid set is used to find the pose ranking which finally define the most distinguishing key-pose. Table 4.2 shows the confusion matrix for the 800 key-poses. We obtained 97.5% recognition rate. We observe that *Jogging* action is confused with *Walking* and *Running* actions. Similarly, *Boxing* action is confused with *Clapping* actions. These confusions are due to the similarity of the corresponding action key-poses. For example, *Jogging* has similar pose variations with *Running* and *Walking*.

4.4.4 Recognition on Weizmann:

For this dataset, also a LOOCV setup is used. We use 200 key-poses as indicated by the optimum pose number for each LOOCV iteration. Table 4.3 demonstrate the confusion matrix. We obtain 98.69% recognition rate. Major confusion observed for the actions: *Jump* - *PJump*, *Skip* - *Run*, *Wave1* - *Wave2* and *Skip* - *PJump* actions. In all these cases there are high similarity among the key-poses.

Table 4.2: Confusion matrix of the KTH dataset. Confusions in % are based on a LOOCV test.

	Walking	Jogging	Running	Boxing	Waving	Clapping
Walking	**98.0**	1.0	1.0	0.0	0.0	0.0
Jogging	3.0	**94.0**	3.0	0.0	0.0	0.0
Running	0.0	2.0	**98.0**	0.0	0.0	0.0
Boxing	0.0	0.0	0.0	**98.3**	0.0	1.7
Waving	0.0	0.0	0.0	0.0	**100.0**	0.0
Clapping	0.0	0.0	0.0	2.0	1.3	**96.7**

Table 4.3: Confusion matrix of the Weizmann dataset. Confusions in % are based on a LOOCV test.

	Jack	Jump	PJump	Run	Side	Skip	Walk	Wave1	Wave2
Jack	**100.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Jump	0.0	**98.0**	2.0	0.0	0.0	0.0	0.0	0.0	0.0
PJump	0.0	1.5	**98.5**	0.0	0.0	0.0	0.0	0.0	0.0
Run	0.0	0.0	0.0	**98.3**	1.0	0.0	0.7	0.0	0.0
Side	0.0	0.0	0.0	0.0	**100.0**	0.0	0.0	0.0	0.0
Skip	0.0	3.4	0.0	0.0	0.0	**96.6**	0.0	0.0	0.0
Walk	0.0	0.0	0.0	1.0	0.0	0.0	**99.0**	0.0	0.0
Wave1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**98.3**	1.7
Wave2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	**99.5**

4.4.5 Recognition on CVC action:

For CVC action also a LOOCV setup is used. Here the optimum key-pose is 500. Table 4.4 shows the confusion matrix. A 98.46% recognition rate is obtained. Compared to other two datasets, we obtain lower confusion between *Waving* - *Two-hand-Wave* actions. Confusion of *Jogging* action is also lower with the *Walking* and *Running* actions. This is due the high resolution of the dataset where directional HOG features are computed more accurately.

4.4.6 Cross-data evaluation

In this part of the experiment we learn the key-pose from one dataset and train the action recognition SVMs. This learning framework is applied to the test dataset. In particular, i) training is done using KTH while Weizmann and CVC action are used as test datasets, ii) Weizmann is used as training while CVC action is used test. We choose this setup to obtain maximum common actions. Confusion matrix for Weizmann-KTH is shown in Table 4.5. We obtain 98.33% overall recognition rate. For CVC-KTH combination, the confusion matrix is presented in Table 4.6, where 98% overall recognition rate is obtained. This this two cases, it worth noting that

Table 4.4: Confusion matrix of the CVC action dataset. Confusions in % are based on a LOOCV test.

	Walking	Jogging	Running	Bending	PJump	Wave1	Wave2
Walking	**98.0**	1.5	0.5	0.0	0.0	0.0	0.0
Jogging	2.0	**96.5**	1.5	0.0	0.0	0.0	0.0
Running	1.0	1.0	**98.0**	0.0	0.0	0.0	0.0
Bending	0.0	0.0	0.0	**99.0**	1.0	0.0	0.0
PJump	0.0	0.0	0.0	0.0	**100.0**	0.0	0.0
Wave1	0.0	0.0	0.0	0.0	0.0	**98.7**	1.3
Wave2	0.0	0.0	0.0	0.0	0.0	1.0	**99.0**

arm based actions are never confused with leg based actions. This is the support of the robust limb dynamics feature and key-pose based action recognition system. For the combination of CVC action-Weizmann dataset, we obtain 97.86% recognition rate. We observe that major confusion between, *Walking-Running*, *Jump-PJump* and *Wave1-Wave2* actions. This is again due the high similarity between key-poses representing those actions.

Table 4.5: Confusion matrix of the Weizmann dataset while KTH is used as training dataset. Confusions in % are based on a LOOCV test.

	Walking	Running	Waving
Walking	**97.0**	3.0	0.0
Running	2.0	**98.0**	0.0
Waving	0.0	0.0	**100.0**

Table 4.6: Confusion matrix of the CVC action dataset while KTH is used as training dataset. Confusions in % are based on a LOOCV test.

	Walking	Running	Jogging	Waving
Walking	**97.5**	1.5	2.0	0.0
Running	1.0	**98.5**	0.5	0.0
Jogging	2.0	2.0	**96.0**	0.0
Waving	0.0	0.0	0.0	**100.0**

4.5 Conclusions

In this Chapter we present a weak model based human action recognition framework by using action key-poses which are obtain by k-medoids clustering on limb dynamics features. We apply directional HOG features to compute limb dynamics as a visual cue. For limb region identification, a ASEF human detector is applied. We introduce a verification human detector using HOG + SVM framework to cope with the

Table 4.7: Confusion matrix of the CVC action dataset while Weizmann dataset is used as training dataset. Confusions in % are based on a LOOCV test.

	Walking	Running	Jumping	Wave1	Wave2	PJump	Bending
Walking	**97.0**	3.0	0.0	0.0	0.0	0.0	0.0
Running	1.5	**98.5**	0.0	0.0	0.0	0.0	0.0
Jumping	0.0	0.0	**97.0**	0.0	0.0	3.0	0.0
Wave1	0.0	0.0	0.0	**97.0**	3.0	0.0	0.0
Wave2	0.0	0.0	0.0	1.0	**99.0**	0.0	0.0
PJump	0.0	0.0	3.5	0.0	0.0	**96.5**	0.0
Bending	0.0	0.0	0.0	0.0	0.0	0.0	**100.0**

scale variation. This is an important extension of the ASEF human detector which overcome the fixed scale problem of ASEF.

We obtain $\sim$ 98% accuracy in KTH, Weizmann and CVC action datasets. We perform a cross-data evaluation, where key-poses are learned from one dataset and used to perform action classification for other dataset. We also obtain state-of-art performance in this case.

Chapter 5

Selective STIP detector for human action detection in complex scenes

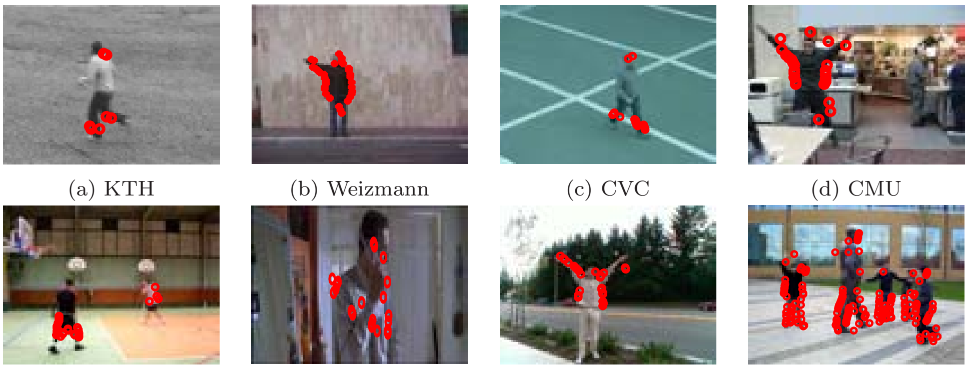
"Everything should be made as simple as possible, but not one bit simpler."

by Albert Einstein.

Recent progress in the field of human action recognition points towards the use of Spatio-Temporal Interest Points (STIPs) for local descriptor-based recognition strategies. In this chapter we present a new approach for STIP detection by applying surround suppression combined with local and temporal constraints. Our method is significantly different from existing STIP detectors and improves the performance by detecting more repeatable, stable and distinctive STIPs for human actors, while suppressing unwanted background STIPs. For action representation we use a bag-of-visual words (BoV) model of local N-jet features to build a vocabulary of visual-words. To this end, we introduce a novel vocabulary building strategy by combining spatial pyramid and vocabulary compression techniques, resulting in improved performance and efficiency. Action class specific Support Vector Machine (SVM) classifiers are trained for categorization of human actions. A comprehensive set of experiments on existing benchmark datasets, and more challenging datasets of complex scenes, validate our approach and show state-of-the-art performance.

In the previous chapters model based approaches for human action recognition is presented, where a detection framework is necessary to identify the action region. Due to the limitation of the this detection algorithms, model based approaches are often lead to low performance for action recognition in complex scenes in diverse and realistic settings (background clutter, camera motion, occlusions and illumination variations).

In this chapter, we explore the concept of spatio-temporal interest points (STIPs) as a tool to capture the local motion pattern for robust action recognition. This idea is inspired by the recent research trend towards the STIP based action recognition techniques [20, 82, 94, 109, 120, 133, 157, 205, 209, 211].



(a) KTH (b) Weizmann (c) CVC (d) CMU

(a) YouTube (b) Hollywood 2 (c) MSR I (d) Multi-KTH

Figure 5.1: Example images with superimposed STIPs from 8 action datasets applied for evaluation of our approach: KTH, Weizmann, CVC, CMU, YouTube, Hollywood 2, MSR I and Multi-KTH. The examples give an indication of the described challenges and differences in the datasets: simple scenes (KTH and Weizmann), semi-complex (CVC), and scenes of high complexity (CMU, YouTube, Hollywood 2, MSR I and Multi-KTH).

In this chapter, we introduce a novel approach for selective STIP detection (different from [44, 102]) by applying surround suppression combined with local and temporal constraints, and achieve robustness to camera motion and background clutter. For action representation we use a BoV model of local N-jet features, extracted at the detected STIPs, to build a vocabulary of visual-words. To this end, we introduce a novel vocabulary building strategy by combining (i) a pyramid structure to capture spatial information and (ii) vocabulary compression. Action class-specific SVM classifiers are trained and applied for categorization of natural human actions. Figure 5.2 shows our selective STIPs detection on the *eight* action recognition datasets used for evaluation of out approach.

The remainder of the chapter is organized as follows. We describe our STIP detector and local descriptor-based action representation in section 5.1. Section 5.2 outlines our vocabulary building strategy and narrates the applied classifier for action categorization. Experimental results and comparisons, along with our technique for spatio-temporal clustering of STIPs for automatic action annotation of Multi-KTH, are reported in section 5.3, followed up by concluding remarks in section 5.4.

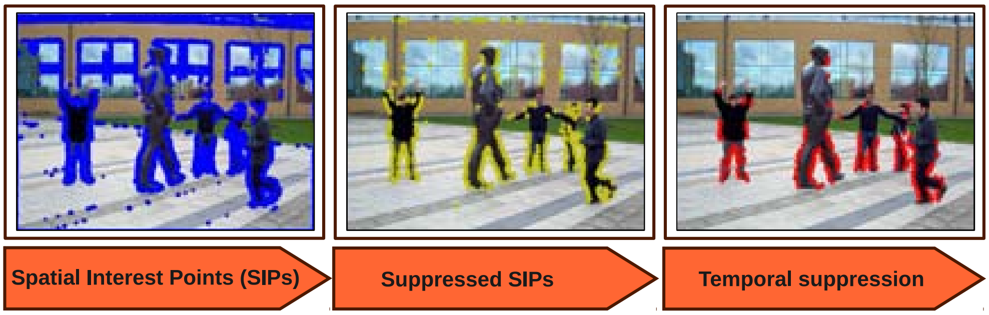


Figure 5.2: A schematic overview of the spatio-temporal interest point detection module and the associated data flow pipeline.

5.1 Selective spatio-temporal interest points

5.1.1 Detection of spatial interest points.

Existing STIP detectors [44, 80, 102, 202, 203] are vulnerable to camera motion and moving background in videos, and therefore detect unwanted STIPs in the background (see Figure 5.3). Cao et al. [25] have recently reported, that of all the STIPs detected by Laptev's STIP detector [102], only about 18% correspond to the three actions performed by the actors in the MSR I dataset [217], while the rest of the STIPs (82%) belong to the background. To overcome this problem, we first detect the spatial interest points (SIPs), then perform background suppression and impose local and temporal constraints (see Figure 5.2). We apply the basic Harris corner detector [70] and compute the first set of interest points C_σ, where σ is the spatial scale. Apart from the detected SIPs on the human actors, the obtained spatial corners C_σ contain a significant amount of unwanted background SIPs (see Figure 5.2).

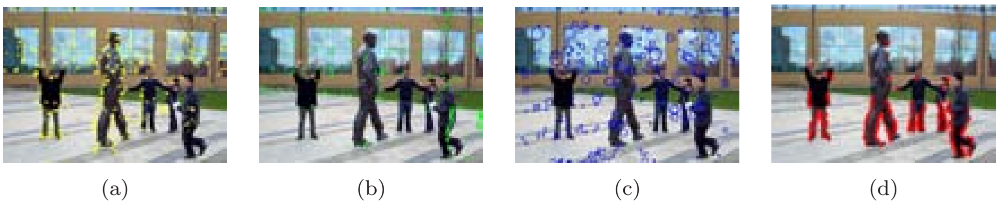


Figure 5.3: STIP detection results for the Multi-KTH dataset. (a) Laptev et al. [102], (b) Dollar et al. [44] (c) Willems et al. [202] and (d) Our approach. Due to background clutter and camera motion (a), (b) and (c) detect quite a large number of STIPs in the background compared to our approach.

5.1.2 Suppressing background interest points

The main idea of our spatial interest point suppression originates in the fact that most corner points detected in the background texture or on non-human objects follow some particular geometric pattern, while those on humans do not have this property. For suppression we use a surround suppression mask (SSM) for each interest point, taking the current point under evaluation as the center of the mask. We then estimate the influence of all surrounding points of the mask on the central point, and accordingly, a suppression decision is taken. The idea is motivated by [65], where surround suppression is used for texture edges to improve object contour and boundary detection in natural scenes. The similar concept of surround suppression based on center surround saliency measure is been adopted in tracking [48], spatio-temporal saliency algorithm [122] and detection of suspicious coincidences in visual recognition [54]. We implement surround suppression by computing an inhibition term for each point of C_σ. For this purpose we introduce a gradient weighting factor $\triangle_{\Theta,\sigma}(x, y, x-u, y-v)$, which is defined as:

$$\begin{gathered}\triangle_{\Theta,\sigma}(x, y, x-u, y-v) = \\ |\cos(\Theta_\sigma(x, y) - \Theta_\sigma(x-u, y-v))|\end{gathered} \tag{5.1}$$

where $\Theta_\sigma(x, y)$ and $\Theta_\sigma(x-u, y-v)$ are the gradients at point (x, y) and $(x-u, y-v)$, respectively; u and v define the horizontal and vertical range of the SSM. If the gradient orientations at point (x, y) and $(x-u, y-v)$ are identical, the weighting factor attains its maximum ($\triangle_{\Theta,\sigma} = 1$), while the value of the factor decreases with the angle difference and reaches a minimum ($\triangle_{\Theta,\sigma} = 0$), when the two gradient orientations are orthogonal. Hence, the surrounding interest points which have the same orientation, as that of (x, y), will have a maximal inhibitory effect.

For each interest point $C_\sigma(x, y)$, we define a suppression term $t_\sigma(x, y)$ as the weighted sum of gradient weights in the suppression surround of that point:

$$\begin{aligned} t_\sigma(x, y) \quad = \quad & \iint_\Omega C_\sigma(x-u, y-v) \\ & \times \triangle_{\Theta,\sigma}\,(x, y, x-u, y-u)dudv \end{aligned} \tag{5.2}$$

where Ω is the image coordinate domain. We now introduce an operator $C_{\alpha,\sigma}(x, y)$, which takes its inputs: the corner magnitude $C_\sigma(x, y)$ and the suppression term $t_\sigma(x, y)$:

$$C_{\alpha,\sigma}(x, y) = H(C_\sigma(x, y) - \alpha t_\sigma(x, y)) \tag{5.3}$$

where $H(z) = z$ when $z \geq 0$ and *zero* for negative z values. The factor α controls the strength of the surround suppression. If no interest points have been detected in the surrounding texture of a given point, the response of the operator retains the original corner magnitude $C_\sigma(x, y)$. However, if a large number of interest points are detected in the surrounding background texture, the suppression term $t_\sigma(x, y)$ will be higher, resulting in a suppression of the current interest point under evaluation.

5.1.3 Imposing local constraints

We select a final set of interest points from the surround suppression responses $C_{\alpha,\sigma}$ (Equation 5.3) by applying non-maxima suppression, similar to Grigorescu et al.'s method for suppressing gradients [65]. Non-maxima suppression thins the areas in which $C_{\alpha,\sigma}$ is non-zero to one-pixel wide candidate contours as follows: for each position (x, y), the two responses $C_{\alpha,\sigma}(x', y')$ and $C_{\alpha,\sigma}(x'', y'')$ in adjacent positions (x', y') and (x'', y''), which are intersection points of a line passing through (x, y) with orientation $\Theta_\sigma(x, y)$ and a square defined by the diagonal points of an 8-neighbourhood, are computed by linear interpolation (see Figure 5.4). A point is kept, if the response $C_{\alpha,\sigma}(x, y)$ is greater than that of the two adjacent points, i.e., it is a local maximum of the neighbourhood. Otherwise its value is set to zero. Figure 5.5 shows an example of the performance of our inhibitive SIP detector. As can be seen in Figure 5.5.b some background SIPs might remain in $C_{\alpha,\sigma}$. However, these static SIPs can be removed by imposing temporal constraints.

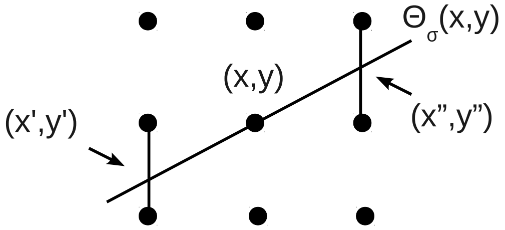


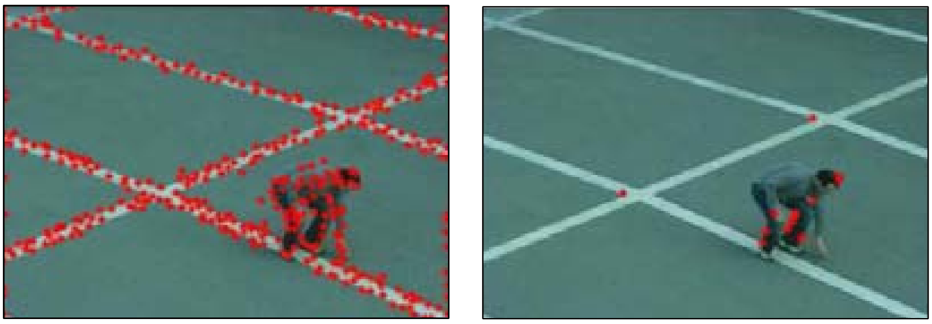
Figure 5.4: Responses at position (x', y') and (x'', y'') along the line passing through (x, y) [65]. Non-maxima suppression retains the value in the central position (x, y), if it is greater than the values at (x', y') and (x'', y'').

5.1.4 Scale adaptive SIPs

Scale selection plays an important role in the detection of spatial interest points. Automatic scale selection can be achieved based on the maximization of normalized derivatives expressed over scale, or by the behavior of entropy or error measures evaluated over scale [22, 110]. Instead of applying an automatic scale selection, as in [100], we apply a multi-scale approach [104] and compute suppressed SIPs in *five* different scales $S_\sigma = \{\frac{\sigma}{4}, \frac{\sigma}{2}, \sigma, 2\sigma, 4\sigma\}$. We follow the idea of scale selection presented by Lindeberg [110] to keep the best set of SIPs obtained for each scale. The best scales are selected by maximizing the normalized differential invariant,

$$\tilde{\kappa}_{norm} = \sigma_0^{2\gamma} L_y L_{xx}. \tag{5.4}$$

Lindeberg [110] report that $\gamma = \frac{7}{8}$ performs well in practice to achieve the maximum value of $(\tilde{\kappa}_{norm})^2$ for spatial interest point detected at multiple scales. After computing the suppressed SIPs in the scale-space in S_σ, we apply this scale selection procedure based on the normalized differential invariant (Equation 5.4), and keep the n best SIPs as our final set of suppressed SIPs.



(a) (b)

Figure 5.5: Performance of our SIP detector with $\alpha = 1.5$. Detected SIPs (a) before suppression and (b) after suppression.

5.1.5 Imposing temporal constraints

After obtaining the final set of spatial interest points we impose temporal constraints to neglect static SIPs. We consider two consecutive frames at a time and remove the common interest points, since static interest points do not contribute any motion information:

$$\mathcal{P}^{T}_{\alpha,\sigma} = C^{T}_{\alpha,\sigma} \backslash \{C^{T}_{\alpha,\sigma} \cap C^{T-1}_{\alpha,\sigma}\} \tag{5.5}$$

where $C^{T}_{\alpha,\sigma}$ is the set of interest points in the T^{th} frame. To avoid the camera motion we have used an interest point matching algorithm as in [88] along with a temporal Gabor filter response to remove the static interest points (Equation 5.5). The remaining points are the final set of detected STIPs, which are used to extract local features. The pseudo code for the full STIP detection is described in Algorithm 5. Parallelization can be adopted for speed optimization by parallel computation of the **for** loops in each algorithm (Algorithm 5,6 & 7).

5.1.6 Local feature descriptors

We use local N-jet features [96] extracted at the detected STIPs. We extract N-jet features of order-2 in five different temporal scales. Consequently, we end up with a 10-dimensional feature vector,

$$\mathcal{F}_{norm}(g(\cdot;\sigma_0,\tau_0)\cdot I) = \{L, \sigma L_x, \sigma L_y, \ldots, \tau^2 L_{tt}\} \tag{5.6}$$

at locally adopted scale level (σ_0, τ_0) for the image sequence I; where $g(\cdot;\sigma_0,\tau_0)$ is the Gaussian kernel at spatio-temporal scale (σ_0,τ_0) and σ_0 is identical to the scale of the STIP detector; $L = g(\cdot;\sigma_0,\tau_0) \otimes I$, i.e. the image I is convoluted with the Gaussian kernel g; L_x is the first order x derivative and L_{xx} is the second order x derivative of L etc.

Algorithm 5 STIP detection from an image stack

```
Require: An image stack (H × W × N), where frames of the videos are saved: $iS$;
    Array containing spatial scales: $sigmaArray$;
    Alpha: $\alpha$;
    Mask: $m$
Ensure: Detected STIPs: $stip$
$N = size(iS, 3)$ (Number frames in the video)
for $i = 1 \rightarrow N$ do
    for $j = 1 \rightarrow size(sigmaArray)$ do
        $SIP(j) = SCD(iS(:, :, i), sigmaArray(j), \alpha, m)$
    end for
    $stip(i) = blobDetector(iS(:, :, i), SIP, sigmaArray)$
end for
$stip = temporalConstraint(stip)$
Return($stip$)
```

These features are computed with a fixed spatial scale σ_0 but with five different temporal scales $(\frac{\tau}{4}, \frac{\tau}{2}, \tau, 2\tau, 4\tau)$. We do not increase the order of N-jet, like Laptev et al. [101], since the two first levels represent velocity L_{xt} and acceleration L_{tt} information, while higher order spatial or temporal derivatives are sensitive to noise and do not bring significant additional motion information. The experimental results reported in section 5.3 document our feature selection by showing state-of-the-art performance.

5.2 Vocabulary building and classification

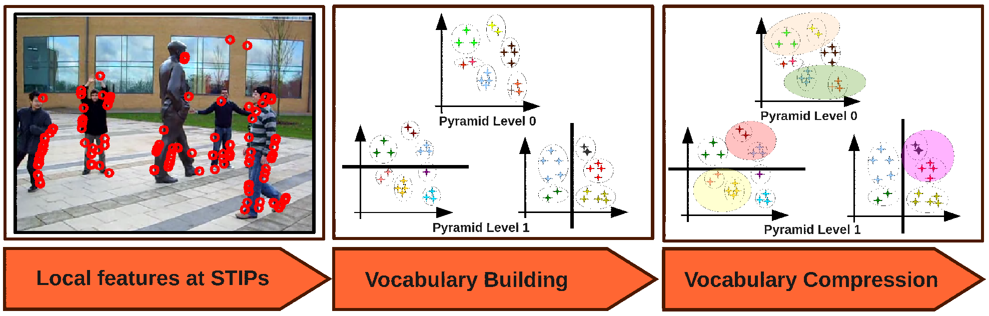


Figure 5.6: A schematic overview of the vocabulary building module and the associated data flow pipeline.

We apply a BoV model to learn the visual vocabularies of the extracted local motion features. We extend the idea of [115] by introducing pyramid levels in the feature space, but instead of applying a pyramid at feature level, as in [113], we apply it at STIP level. This makes the problem of grouping the local features much simpler

Algorithm 6 SCD: Selective STIP detection

```
Require: An image (H × W): $image$;
    Spatial scale: $\sigma$;
    Alpha: $\alpha$;
    Mask: $mask$
Ensure: Detected selective spatial interest points: $sip$
$cp = harrisCorner(image, \sigma)$
$\Theta = gradient(image)$
for Each point $(x, y) \in cp$ do
    $\triangle_{\Theta_{mask}} = |\cos(\Theta_{mask} - \Theta_{mask_{(x,y)}}|$
    $t(x, y) = cp_{mask} \otimes \triangle_{\Theta_{mask}}$
    $cp(x, y) = H(cp_{(x,y)} - \alpha t_{(x,y)})$
    $(x', y') = line(x + 1, y, \Theta(x, y))$
    $(x'', y'') = line(x - 1, y, \Theta(x, y))$
    if $(cp(x, y) > cp(x', y')) \wedge (cp(x, y) > cp(x'', y''))$ then
        Keep $cp(x, y)$
    end if
end for
Return($cp$)
```

yet robust, since our STIPs are detected in a selective and robust manner. Finally, we apply vocabulary compression, at each pyramid level, to reduce the dimensionality of the feature space (see Figure 5.6).

5.2.1 Pyramid structure

Let I_T be the T^{th} frame of the image sequence I and $P^T_{\alpha,\sigma}$ (Equation 5.5) the set of detected STIPs in this frame. We then quantize this set of STIPs into q levels, $\mathcal{S} = \{s_0, s_1, \ldots, s_{q-1}\}$ [125]. For each of these levels, the STIPs are divided based on center of mass information. Accordingly, we group the motion features into different levels of the pyramid. The structure of our 2-level pyramid is illustrated in Figure

Algorithm 7 temporalConstraint: Impose temporal constraint on the selected spatial corner points

```
Require: An image stack (H × W × N): $iS$;
    Spatial corner points: $cp$
Ensure: Detected STIPs: $stip$
for $i = N \rightarrow 2$ do
    $f_1 = iS(:, :, i)$
    $f_2 = iS(:, :, i - 1)$
    $cp(i) = cp_{f_1} - pointMatch(cp_{f_1}, cp_{f_2})$
end for
Return($cp$)
```

5.7. The horizontal division helps to capture the distinguishing characteristics of arm and leg-based actions, whereas the vertical division distinguishes the actions within each of these arm and leg-based action classes.

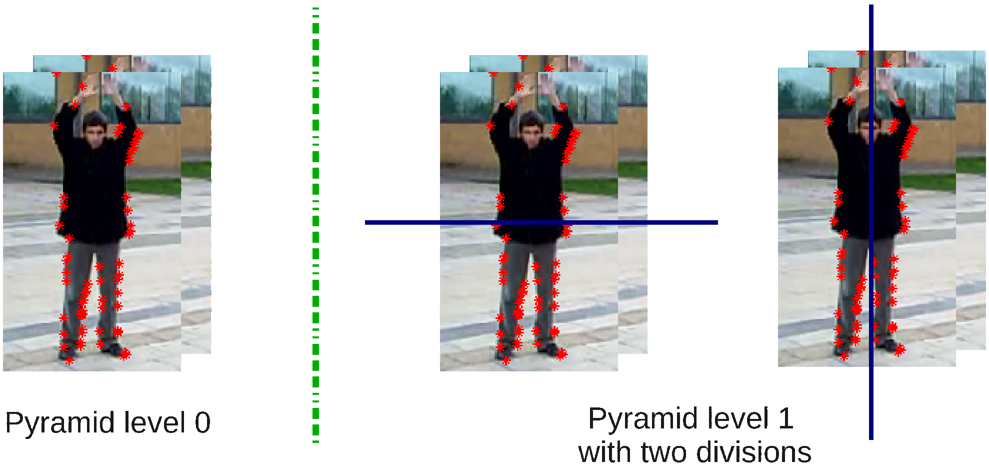


Figure 5.7: Spatial pyramid of level 2.

5.2.2 Vocabulary compression

After dividing the motion features into the described pyramid levels, we create initial vocabularies of a relatively large size (about 400 words). To reduce the final feature dimensionality, we use vocabulary compression, as in [115], but at each level of the pyramid to achieve a compact yet discriminative visual-word representation of actions.

Let A be a discrete random variable which takes the value of a set of action classes $A = \{a_1, a_2, \ldots, a_n\}$, and W_s be a random variable which range over the set of video-words $W_s = \{w_1, w_2, \ldots, w_m\}$ at pyramid level s. Then the information about A captured by W_s can be expressed by the Mutual Information (MI), $I(A, W_s)$. Now, let $\widehat{W_s} = \{\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_k\}$ for $k < m$, be the compressed video-word cluster of W_s. We can measure the loss of quality of the resulting compressed vocabulary $\widehat{W_s}$, as the loss of MI:

$$Q(\widehat{W_s}) = I(A, W_s) - I(A, \widehat{W_s}) \tag{5.7}$$

To find the optimal compression $\widehat{W_s}$ we use an Agglomerative Information Bottleneck (AIB) approach.

5.2.3 AIB compression

AIB [170] iteratively compresses the vocabulary W_s by merging the visual-words w_i and w_j which cause the smallest decrease in MI, $I(A, W_s)$. The algorithm can be summarized as follows:

- Initiate $\widehat{W_s} \equiv W_s$, i.e., by taking each video-word of W_s as a singleton cluster.
- Pair-wise distance computation: for every $\{w_i, w_j\} \in \widehat{W_s}$, $i < j$, the distance d_{ij} (which is a measure of MI) is computed:

  $$d_{ij} = (p(w_i) + p(w_j)) \cdot JS_{\Pi}[p(a|w_i), p(a|w_j)] \tag{5.8}$$

  where $JS_{\Pi}[p(a|w_i), p(a|w_j)]$ is the Jensen-Shannon divergence for a M class distribution, $p_i(x)$, each with a prior π_i, and is defined as:

  $$JS_{\Pi}[p_1, p_2, \ldots, p_M] \equiv H[\sum_{i=1}^{M} \pi_i p_i(x)] - \sum_{i=1}^{M} \pi_i H[p_i(x)] \tag{5.9}$$

  where $H[p(x)]$ is Shannon's entropy:

  $$H[p(x)] = -\sum_{x} p(x) \log p(x) \tag{5.10}$$

- Merging: select the pair of video-words $\{w_\alpha, w_\beta\}$ for which the distance $d_{\alpha\beta}$ is minimum and merge them. Hence, we merge the video-words which result in the minimum MI loss by optimizing the global criterion in Equation 5.7.

AIB is a greedy algorithm in nature and optimizes the merging of only two word clusters at every step (local optimization). Hence, it optimizes the global criteria defined in Equation 5.7. We use the described vocabulary compression at each level of the pyramid per class, and obtain a final class-specific compact pyramid representation of video-words.

We use AIB for the vocabulary compression instead of Principal Component Analysis (PCA) based dimensionality reduction, since PCA is a linear model, whereas the relationship among the video words are highly non-linear in nature. Besides, PCA based dimensionality reduction will work on the first level cluster (k-means) of the bag-of-words model to reduce the final bag-of-words histogram dimensionality. Hence, it will not take inter and intra cluster similarities into account. Unlike PCA, the agglomerative information bottleneck (AIB) method presented in the article, is non-linear and it yields a set of compressed clusters from the first level clusters, such that the set of resulting compressed clusters maximally preserves the original information among them. Additionally, AIB based compression explores the mutual information present among video words and apply compression based on this information. Hence, in this case, AIB based compression is analytically more appropriate than PCA.

To empirically support our selection of AIB based compression, we have conducted experiments on the Weizmann dataset using PCA based dimensionality reduction. The obtained average accuracy is quite low (40% in the range of $30\% - 70\%$ compression) compared to the recognition rate of AIB (99% in the same range of compression), which documents that AIB is a far better choice.

Table 5.1: Average recognition accuracy for the Weizmann dataset using different SVM kernels. We have used a Polynomial kernel of degree 3.

SVM Kernel	Recognition rate (%)
χ-square	**99.50**
Intersection	97.78
Radial basis function	87.77
Polynomial	78.67
Linear	58.89

5.2.4 Action classification

After compression of the video-words at each pyramid level we compute a histograms of the video-words, using the extracted local motion features, and concatenate them to a final feature set for SVM learning. We design a class specific χ-square kernel-based SVM, $\text{SVM}_{a_i}(k, h^{a_i}_{W_{a_i}})$ [31][1], where a_i is the i^{th} action class A, k is the SVM kernel and $h^{a_i}_{W_{a_i}}$ is the histogram of action class a_i, computed using the class-specific video-words W_{a_i}. For a test set a_{Test} we detect its action class:

$$i^*_{a_{Test}} = argmax_j \text{SVM}_{a_j}(k, h^{a_{Test}}_{W_{a_j}}), \forall a_j \in A \tag{5.11}$$

We conduct experiments using different SVM kernels, and observe that the χ-square and intersection kernel are the best perfoming SVM kernels for all the datasets. Hence, we apply the χ-square kernel for all our experiments on human action recognition in section 5.3. Table 5.1 shows the average recognition accuracy for the Weizmann dataset using a number of different SVM kernels.

5.3 Experimental results

5.3.1 Human action datasets

To test our proposed approach for action recognition we conduct a comprehensive set of experiments using a number of publicly available human action datasets , which are categorized as follows.

- **Single actor benchmark:** To conduct benchmark testing we choose the two most popular human action datasets: KTH [159] and Weizmann [63].
- **Single actor with complex background** In this category we choose the CVC action dataset [2] and the CMU action dataset [91].
- **Movie and YouTube video clips** To evaluate our approach in different challenging stettings, we conduct experiments on movie and YouTube video clips.

[1] http://www.csie.ntu.edu.tw/∼cjlin/libsvm

[2] http://iselab.cvc.uab.es/files/Tools/Cvc-ActionDataSet/index.htm

Concretely, we use the Hollywood 2 human actions and scenes dataset [125] and the YouTube action dataset [115].

- **Multiple actors with complex background** We use two multiple actor datasets: the Microsoft research action dataset I (MSR I) [217] and the Multi-KTH dataset [183].

Please refer to Chapter A for detail descriptions of these datasets.

5.3.2 Automatic action annotation for Multi-KTH

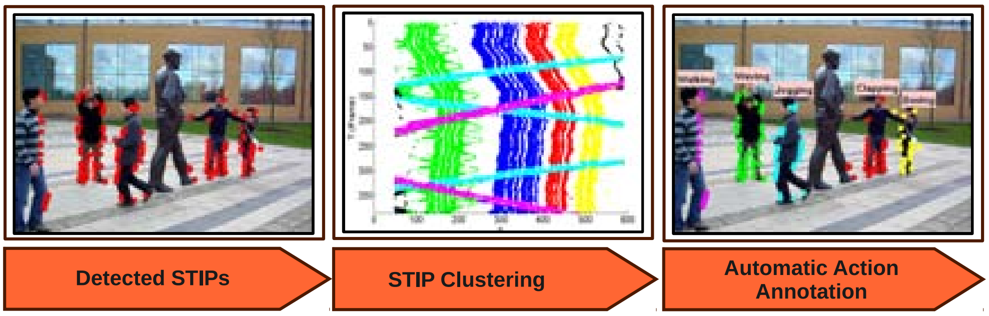
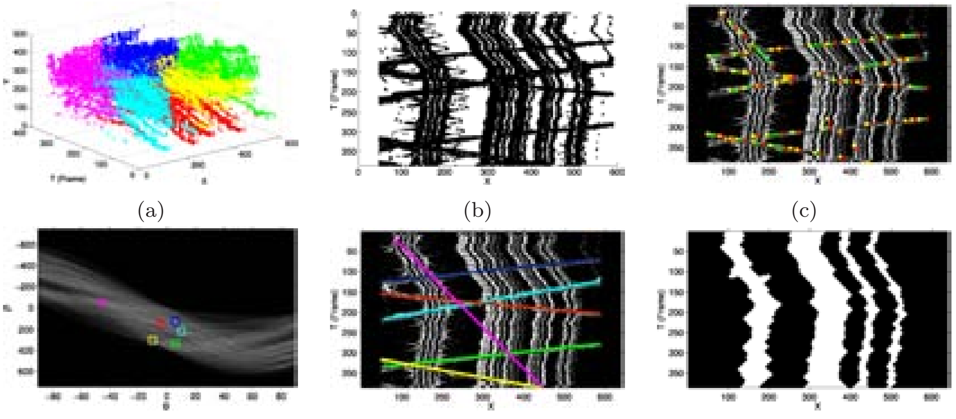


Figure 5.8: A schematic overview of the spatio-temporal clustering module and the associated data flow pipeline.



(a) (b) (c)

(d) (e) (f)

Figure 5.9: Plots of the detected STIPs for the Multi-KTH dataset, and detection of linear patterns in the XT-space. (a) k-means clustered STIPs in the $3D$ spatio-temporal XYT-space and (b) ungrouped STIPs in the $2D$ spatio-temporal XT-space; (c) line segments in XT-space caused by actions like walking, jogging or running; (d) candidates with high responses in the Hough space; (e) detected line segment using the Hough transfrom and (f) *blobs* obtained by morphological operations.

When multiple actors appear simultaneously in a scene, it is necessary to group the detected STIPs into actor-specific clusters. An excellent example is the Multi-KTH dataset, where five actors are present in the scene. Based on this dataset we introduce a spatio-temporal clustering technique for actor-specific STIP grouping and evaluate its performance in section 5.3.8. This spatio-temporal clustering is only a part of Multi-KTH dataset for automatic annotation.

Actor-specific STIP clustering

The actions present in the Multi-KTH dataset can be divided into two main groups: the actions with moving actors, like *walking* and *jogging*, and the actions with static actors, like *boxing*, *waving* and *clapping.* These two different nature of actions can be analyzed in the 2D spatio-temporal XT-space (see Figure 5.9.b). The actor-specific STIP clustering exploits the 2D spatio-temporal XT-space and consist of two main steps:

i) detection of lines in the XT-space and cluster STIPs accordingly,

ii) after the first set of STIP clusters have been estimated, the associated STIPs are excluded and the resulting subset is clustered using morphological operations and a spatio-temporal distance measurement.

The surround suppression effect of our STIP detector, resulting in a low detection rate of unwanted background STIPs, facilitates STIP clustering in the XT-space. This will simply not be possible with a high number of background STIPs. Figure 5.8 illustrates the concept of the spatio-temporal clustering.

The spatio-temporal XT-space

A plot of the detected STIPs in 3D spatio-temporal XYT-space for the Multi-KTH sequence is shown in Figure 5.9.a. As can be seen, actor-specific clustering of the STIPs is non-trivial due to camera motion and occlusions. Hence, successful clustering cannot be accomplished by commonly used methods, e.g., k-means or Mean Shift clustering. Instead, we project the 3D spatio-temporal STIPs onto a 2D spatio-temporal XT-space, as shown in Figure 5.9.b, which reveals some interesting and useful patterns. The XT-space can be seen as the top-down view of the 3D spatio-temporal XYT-space (Figure 5.9.a), with the horizontal and vertical axes representing the X-position and the time T, respectively. Hence, the T-axis demonstrates the evolution of STIPs in time.

Detection of lines in XT-space

Actions like *walking*, *jogging* or *running* create lines in the XT-space. Hence, we detect line segments in XT-space to cluster STIPs detected for the actors. This is

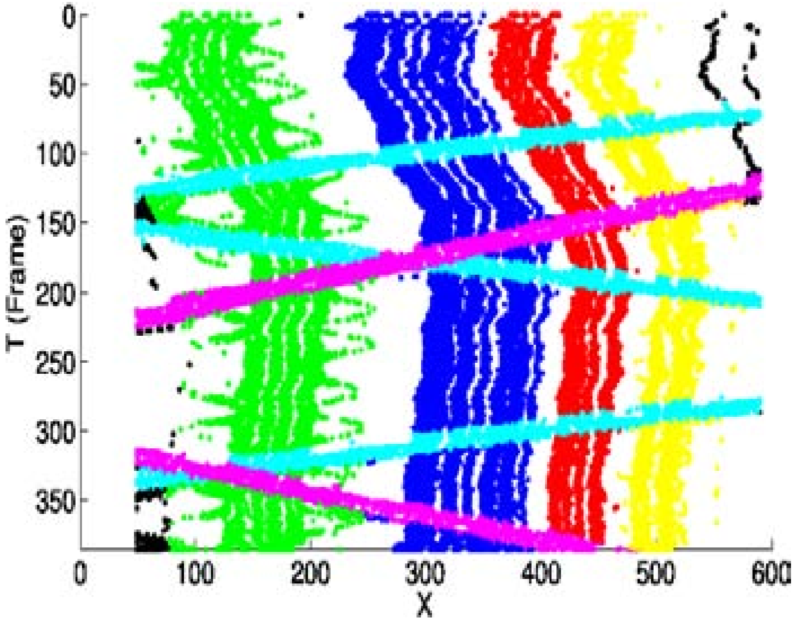
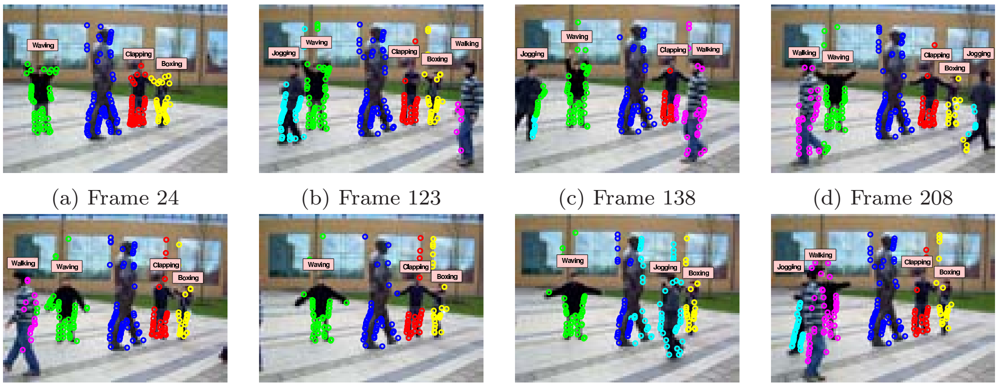


Figure 5.10: Actor-specific STIP clustering in the XT-space.



(a) Frame 24

(b) Frame 123

(c) Frame 138

(d) Frame 208

(e) Frame 217

(f) Frame 234

(g) Frame 296

(h) Frame 333

Figure 5.11: Automatic annotation of STIPs detected for multiple simultaneously actors for a number of frames from the Multi KTH dataset.

valid, since actors with a certain target destination move in a linear pattern for those actions. Hough transform [45] is applied for the detection of these linear patterns (i.e., line segments) and the candidates with high response in the Hough Space are kept. Furthermore, a post candidate approval is applied based on the slope of the lines. Figure 5.9 shows this process and the intermediate results. As can been seen, the erroneously detected (magenta colored) line can be discarded according to its steep slope. Furthermore, Line segments for the crossing actors are detected but due to a high amount of camera motion, it is not possible to detect good candidates for the other actors performing upper body acations, like *boxing*, *clapping* and *waving*.

STIP clustering in XT-space

We use the detected lines to cluster the STIPs by applying a point-line distance measure $d(x, t)$, and threshold according to a maximum distance d_{max} for each line segment:

$$d(x,t) = \frac{|(\mathbf{p} - \mathbf{q_1}) \times (\mathbf{p} - \mathbf{q_2})|}{|\mathbf{q_2} - \mathbf{q_1}|} < d_{max} \tag{5.12}$$

where $\mathbf{p}$ is the current STIP under evaluation, and $\mathbf{q_1}$ and $\mathbf{q_2}$ are two points lying on a detected line. The maximum distance d_{max} is set according to the size of the actors appearing in the dataset. After clustering the first set of STIPs, we exclude them and use the remaining STIPs for further clustering. We merge the new subset of STIPs by morphological operations (see Figure 5.9.f) and use the resulting *blobs* to cluster the STIPs, by considering the spatio-temporal distance between a STIP and the contours. Figure 5.10 shows the resulting actor-specific STIP clustering in the XT-space, and in figure 5.11 the grouped STIPs are superimposed on a number of frames from the Multi-KTH dataset.

Table 5.2: STIP detection ratios (%): the number of STIPs detected on the actors with respect to the total number of detected STIPs, estimated for the MSR I and Multi-KTH datasets using our approach and state-of-the-art methods.

Method	MSR I	Multi-KTH
Our approach	**76.21**	**90.34**
Laptev el al. [102]	18.73	48.16
Dollár et al. [44]	21.36	16.03
Willems et al. [202]	24.02	20.24

5.3.3 Evaluation of STIP detector

We evaluate our STIP detector by estimating a score for the number of detected STIPs for the actors in comparison to those detected in the background. Cao et al. [25] have recently reported that of all the STIPs detected by Laptev's STIP detector [102], only 18.73% correspond to the three actions performed by the actors in the MSR I, while the rest of the STIPs (81.27%) belong to the background. Ground truth bounding boxes are used to determine if a STIP belongs to an action instance. We evaluate our STIP detector on MSR I in a similar way, and detect **76.21**% STIPs for the actors. We observe that our detector tends to detect more points in the background, when applied to the sequences of MSR I with several moving people in the background. Our STIP detector is designed to detect interest point for people, hence it will also

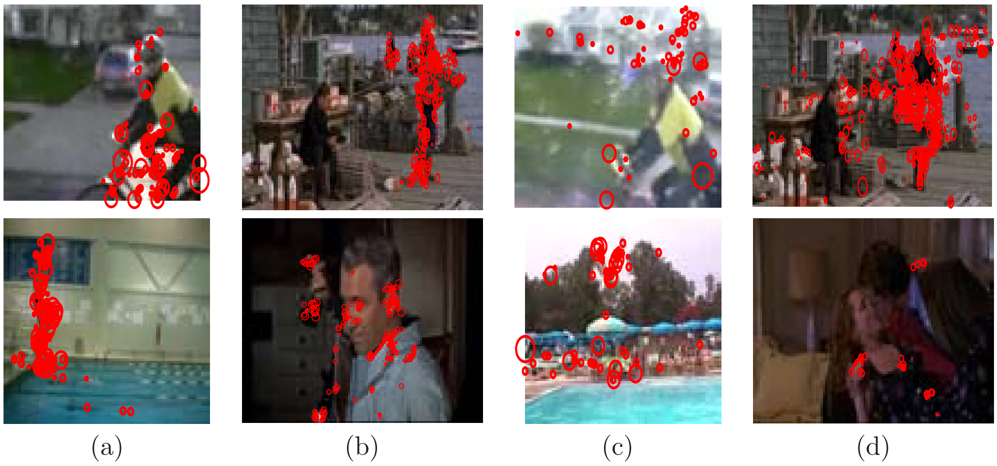


Figure 5.12: Performance of the STIP detector in sequences with complex scenarios. Successful STIP detection is shown for frames of the (a) YouTube and (b) Hollywood 2 dataset, respectively. Additionally, the failure frames of (c) YouTube and (d) Hollywood 2 are also shown. In (a) and (b) our STIP detector successfully handles camera motion and the STIPs are detected only in the motion of interest. On the contrary, in the frames of (c) and (d), due to high background motion and difference in scene resolution the STIP detector loses the focus on the motion of the human actors.

consider moving people in the background as candidates. We also conduct this experiment for the Multi-KTH dataset by manually annotating ground truth bounding boxes, and find that **89.35%** STIPs belong to the actors (see Figure 5.3). This is consistent with the concept of our STIP detector, and documents the effectiveness of our incorporated surround suppression followed up by imposing local and temporal constraints. Table 5.2 shows STIP detection ratios of the state-of-the-art methods, and clearly documents the superior performance of our STIP detector.

The time complexity of our STIP computation highly depends on the size of the input video. For a video of size (160 × 120 × 550), the STIP computation, executed on a standard dual core Desktop PC (Intel(R) Core(TM)2 CPU 6400@2.13GHz 6GB RAM) using MATLAB R2010, takes approximately 10 mins.

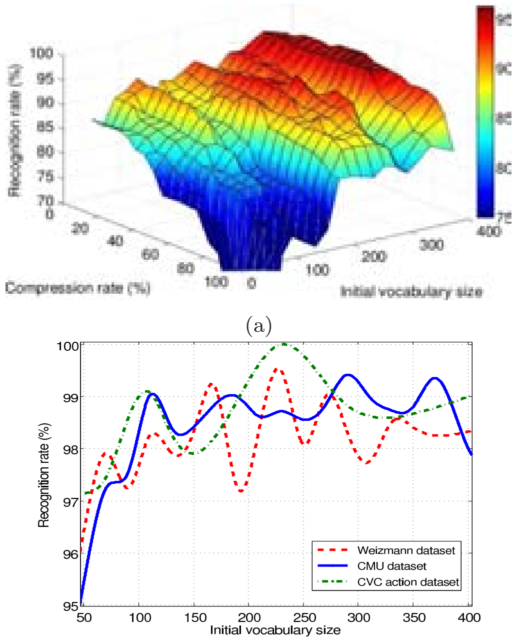
Figure 5.12 shows the perfomance of the STIP detector in complex scenarios. Despite of the camera movement, the STIP detector performs well (Figure 5.12(a) and (b)). However, in some cases, due to the combination of complex backgorund, low resolution and large background motion, the STIP detector loses focus and detects a larger number of background STIPs (Figure 5.12(c)) or an insufficient number of actor STIPs (Figure 5.12(d)).

Table 5.3: State-of-the-art recognition accuracies (%) for the KTH, Weizmann and YouTube datasets. *Liu et al. [115] test on 8 out of the 11 YouTube actions.

Method	KTH	Weiz.	YouTube
Our approach	**96.35**	99.50	**86.98**
Lui et al. [120]	96.00	-	-
Yu et al. [216]	95.67	-	-
Kim et al. [94]	95.33	-	-
Wu et al. [205]	95.10	98.90	-
Cao et al. [25]	95.02	-	-
Kaâniche et al. [89]	94.67	-	-
Kovashka et al. [98]	94.53	-	-
Gilbert et al. [58]	94.50	-	-
Sadek et al. [154]	94.30	-	-
Liu & Shah [113]	94.16	-	-
Sun et al. [176]	94.00	97.80	-
Saghafi et al. [155]	93.94	-	-
Shao et al. [161]	93.89	-	-
Liu et al. [115]	93.80	-	71.20
Uemura et al. [183]	93.70	-	-
Lin et al. [108]	93.43	**100.00**	-
Yuan et al. [217]	93.30	-	-
Liu et al. [115]	92.30	-	76.10*
Yao et al. [211]	93.00	92.20	-
Schindler et al. [157]	92.70	**100.00**	-
Laptev et al. [101]	91.80	-	-
Jhuang et al. [80]	91.70	98.80	-
kläser et al. [95]	91.40	84.30	-
Yang et al. [210]	87.30	99.40	-
Wong et al. [203]	86.62	-	-
Willems et al. [202]	84.26	-	-
Niebles et al. [134]	81.50	-	-
Dollár et al. [44]	81.17	-	-
Schüldt et al. [159]	71.72	-	-
Gorelick et al. [63]	-	99.64	-
Thurau et al. [179]	-	94.40	-
Ali et al. [6]	-	92.60	-
bregonzio et al. [20]	-	-	64.00

5.3.4 Vocabulary building

The purpose of this experiment is to reveal the optimal initial vocabulary size and compression rate for our vocabulary building strategy. We divide each dataset into 50% training, 20% validation and 30% testing partitions. The final training of the SVMs uses both the training and validation sets. The recognition rates are computed



(a)

(b)

Figure 5.13: Revealing the influence of the vocabulary size and compression on the average action recognition rates. (a) A 3D Plot of the recognition rate, as a function of the initial vocabulary size and the compression rate, for the KTH dataset. (b) Recognition rates, as a function of the initial vocabulary size, for the three single actor datasets: CMU, CVC and Weizmann. The compression rate is fixed to 65%, i.e., 35% of the initial vocabulary size is used.

by averaging over 50 random instances of these sets. We conduct experiments using a similar vocabulary size range as Liu et al. [115], with an initial vocabulary size of 50 video-words and incrementing it up to 400. We weight the initial vocabulary size according to the pyramid level using a weight factor 2^{-s}, where s is the pyramid level. The vocabulary size is weighted to avoid the empty/singleton cluster creation in finer levels of the pyramid. We reduce the dimensionality of the final feature vectors for the SVM classifiers by applying vocabulary compression at each pyramid level. To choose the optimal vocabulary size and compression rate, we vary the initial vocabulary size range [50-400] with an increment of 20, and for each of these vocabularies we vary the compression rate from 0% to 95% with an increment of 5%. Figure 5.13.a shows the resulting 3D plot of the recognition rate as a function of the initial vocabulary size and the compression rate, for the KTH dataset. The maximum recognition rate indicates the optimal vocabulary size and compression rate. We observe that the best result is obtained at a compression rate upto **65**%, and the performance starts to degrade rapidly above **80**%. In Figure 5.13.b the recognition rate, as a function of the initial vocabulary size for the three other single actor datasets: CMU, CVC and

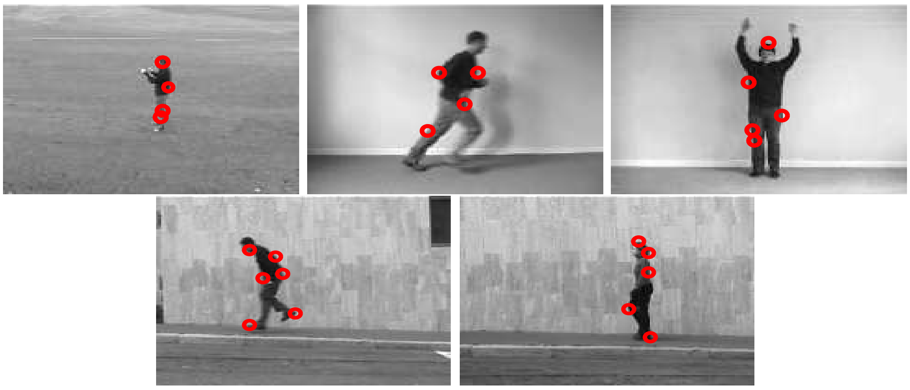


Figure 5.14: Error-frames of the videos that are miss-classified for the KTH and Weizmann datasets shown in first and second rows respectively. Three frames in the first row depict miss-classified *boxing*, *running* and *waving* actions from the KTH dataset, respectively. The two error-frames shown in the last frame are for *skip* and *walking* actions of the Weizmann dataset respectively. These frames show cases which result in miss-classification. Due to low resolution only a limited number of STIPs are detected for the important body parts (arms and legs), which are taking major part in these actions.

Weizmann, is shown. We obtain approximately 100% recognition rate in the initial vocabulary size range [230-300] for the Weizmann, CMU and CVC datasets, which is similar to the middle peak in Figure 5.13.a for KTH.

5.3.5 Benchmark testing

We use the KTH and Weizmann datasets for benchmark testing, and achieve an accuracy of 96.35% for KTH and 99.50% for Weizmann. Table 5.3 shows a comparison of the recognition rates of our approach and several other state-of-the-art methods for these two datasets. It should be noted that we achieve an approximately 2% increase in the recognition rate for KTH. We obtained this perfect recognition with an initial vocabulary size of 270 and a 65% compression rate. The main reasons for this improvement are the selective STIP detection and the spatial pyramids, which capture the local characteristics of actions, and thereby reduce interclass confusion. The accuracy for Weizmann is approximately 100%, which is comparable to the state-of-the-art. Lin et al. [108] report a clear 100% recognition rate for Weizmann. However, this work applies a template matching technique, using holistic features extracted from global boundary box-based interest regions. Furthermore, it requires background subtraction and target tracking. In contrast, our approach uses local features and does not require any preprocessing. Since, Weizmann is a simple datasets without any further challenges, it favors global and holistic methods. In contrast, our approach is applicable for all types of scenes, including very challenging scenes of high complexity, which we will validate in the following.

We analyze the error-frames of the 0.50% videos of the Weizmann dataset, which are miss-classified. Similarly, we analyse the miss-classified frames from the confusion matrix of for KTH. Figure 5.14 shows some example error-frames. Due to low resolution only a limited number of STIPs are detected for the important body parts (arms and legs), which are taking major part in actions like *boxing* and *running*. In these few cases this results in miss-classification.

Table 5.4: The average precision (%) and mean average precision (MAP) for the actions of Hollywood 2, using our apporach in comparison to the state-of-the-art.

Action	Marszalek [125]	Han [67]	Wang [190]	Gilbert [58]	Ullah [184]	**Ours**
AnswerPhone	13.10	15.57	-	40.20	26.30	**41.60**
DriveCar	81.00	87.01	-	75.00	86.50	**88.30**
Eat	30.60	50.93	-	51.50	59.20	**56.50**
FightPerson	62.50	73.08	-	77.10	76.20	**78.20**
GetOutCar	8.60	27.19	-	45.60	45.70	**44.80**
HandShake	19.10	17.17	-	28.90	49.70	**51.60**
HugPerson	17.00	27.22	-	49.40	45.40	**50.30**
Kiss	57.60	42.91	-	56.60	59.00	**57.25**
Run	55.50	66.94	-	47.50	72.00	**74.35**
SitDown	30.00	41.61	-	62.00	62.40	**61.15**
SitUp	17.80	7.19	-	26.80	27.50	**30.00**
StandUp	33.50	48.61	-	50.70	58.80	**60.00**
MAP results	35.50	42.12	47.70	50.90	55.70	**57.83**

Table 5.5: Recognition accuracies (%) for cross-data evaluation trained on KTH and tested on other datasets: Weizmann, CVC, CMU, MSR I and Multi-KTH. The first row presents results when training and testing on the same dataset for Weizmann, CVC and CMU.

Method	Weizmann	CVC	CMU	MSR I	Multi-KTH
Our approach (without cross-data)	**99.50**	**100.00**	**99.42**	-	-
Our approach	**100.0**	**96.95**	**91.94**	**84.77**	**98.40**
Yuan et al. [217]	-	-	70.00	-	-
Cao et al. [25]	-	-	-	60.00	-
Gilbert et al. [59]	-	-	-	-	75.20
Gilbert et al. [58]	-	-	-	-	68.80
Uemura et al. [183]	-	-	-	-	65.40

5.3.6 Evaluation on complex scene

The main objective of this evaluation is to test the capability of our method to handle background clutter. For this purpose we choose the CMU action dataset and the CVC Action dataset with textured background. Despite the presence of strong background texture and clutter, we achieve a 100.0% accuracy rate for CVC and 99.42% for CMU (see Table 5.5). The high performance for both of these dataset is consistence with the theoretical foundation of our proposed STIP detector. The detector's selective behavior, achieved by incorporating surround suppression and imposing local and temporal constraints, results in robustness to background texture and clutter.

5.3.7 Action recognition in movie and YouTube video clips

Next, we conduct experiments on movie and YouTube video clips, using the YouTube and Hollywood 2 action datasets. We achieve 86.98% recognition rate for the YouTube actions. Table 5.3 shows the comparison with other state-of-the-art method for this dataset. Our approach is far superior compared to the other reported methods, due to our STIP detector's capability to handle complex and challenging scenes with camera motion, cluttered background, and variation in scale, viewpoint and illumination.

For the Hollywood 2 dataset, the performance is evaluated as suggested in [125], i.e., by computing the average precision (AP) for each of the action classes and reporting the mean AP over all classes (MAP). Table 5.4 shows the AP for the actions in comparison to other state-of-the-art methods. The Hollywood 2 dataset contains very complex scenes from movies with no ground truth information available, and moreover the different instances of an action are sometimes viewed from different camera angles.

Notes: "*Answerphone* and *Handshake* are quite small, and therefore need a very complex set of compound features in order to classify the action over the background noise. In contrast, *FightPerson* and *DriveCar* use more global contextual features and therefore they work with lower level features."

5.3.8 Cross-data experiments

We perform exhaustive cross-data evaluation to test our proposed method in more realistic scenarios and use the KTH and Weizmann datasets for training data. We observe that the Weizmann dataset is not appropriate for training, and results in a poor 40% and 45% recognition rate for CVC and CMU, respectively. This is due to inadequate training data since Weizmann contains a very limited number of action instances per category compared to KTH. Table 5.5 shows the accuracy rates obtained using KTH as training. These cross-data results validate that our approach is applicable for more practical scenarios, where training and test data are coming from different sources.

The KTH dataset has only one common action, *two-hands-waving*, with the CMU action dataset. We use the KTH *running* sequence as negative data and obtain a 91.94% recognition rate. It is noticeable, that the accuracy is actually higher for Weizmann (100%) and CMU (99.42%), than when training and testing on the same dataset, due to the sufficient action instances for training. Additionally, for CMU we only recognize one action, *two-hands-waving*, compared to five actions when both training and testing on CMU. On the contrary, the accuracy decreases by 3% for CVC, due to its lower inter-dataset correlation with KTH. For the Multi-KTH dataset we manually annotate the action labels as ground truth, using bounding boxes, and obtain 98.40% accuracy. We perform another test using our automatic action annotation described in section 5.3.2, and obtain a 94.20% recognition rate, which is comparable to the results of the manual annotation. For the MSR I dataset we achieve 84.77% accuracy. The difficult part of MSR I is that some sequences contain moving people in the background depicted by the bounding box of the agent performing the action, which result in unwanted STIP in the background, and thereby a lower recognition rate compared to the other datasets. In conclusion, these results outperform the state-of-the-art significantly (see Table 5.5) and hereby validate the robustness of our method in more realistic action recognition scenarios. Although these datasets are very complex and contain several practical challenges: cluttered and moving backgrounds (including people and vehicles), camera motion and multiple actors, our approach performs robustly.

5.4 Conclusion

In this paper we have presented a novel approach for human action recognition in complex scenes. Our approach is based on selective STIPs which are detected by suppressing background SIPs and imposing local and temporal constraints, resulting in more robust STIPs for actors and less unwanted background STIPs. We apply a BoV model of local N-jet descriptors extracted at the detected STIPs and introduce a novel vocabulary building strategy by combining a spatial pyramid and vocabulary compression. Action class-specific SVM classifiers are trained to finally identify human actions.

The strong aspect of our proposed STIP detection method is, it can detect dense STIPs at the motion region without affected by the complex background. This is an important property to detect actions in complex scenarios. Regarding the weak aspect, our method suffers in the presence of other motion (presence of multiple actors) together with the region of action. In this scenario we detect several STIPs from different motion region results in poor classification.

In the current system, we use greedy approach for vocabulary compression. Sometimes, the time complexity is higher with this approach. A non-greedy method for vocabulary compression might be an interesting inclusion for the future work. Our automatic action annotation using STIP clustering works well for the multi-KTH dataset, yet it is not generalized for other multi-actor action datasets. The automatic action annotation for multi-actor datasets is a very difficult and challenging task. We could include more complex shape matching algorithm along with a human model in the XT-space to minimize the overlap in the STIP clusters of the moving and non-moving actors.

We have reported superior action recognition results (∼100% accuracy) in comparison to the state-of-the-art, when testing on benchmark datasets of simple scenes (KTH and Weizmann), and similar performance for complex scenes (CVC and CMU). Note that we have raised the recognition rate for KTH by approximately 5%. Additionally, we have shown state-of-the-art performance and proven the applicability of our approach for action recognition in movie and YouTube video clips by significantly outperforming other methods evaluated on the YouTube action dataset, and showing the highest mean average precision for the Hollywood 2 dataset. A comprehensive cross-data evaluation has been performed by separating the training (KTH) and test datasets (CVC, CMU, MSR I and Multi-KTH). To our best knowledge we are the first to report exhaustive cross-data evaluation. Compared to state-of-the-art we have reported superior results by raising the recognition rates from approximately 60-75% to 85-100%.

Chapter 6

4D STIPs for human action recognition in multi-camera setup

"Somewhere, something incredible is waiting to be known."

by Dr. Carl Sagan.

In this chapter we address the problem of human action recognition in reconstructed 3-dimensional data acquired by multi-camera systems. We contribute to this field by introducing a novel 3D action recognition approach based on detection of 4D (3D space + time) Spatio-Temporal Interest Points (STIPs) and local description of 3D motion features. STIPs are detected in multi-view images and extended to 4D using 3D reconstructions of the actors and pixel-to-vertex correspondences of the multi-camera setup. Local 3D motion descriptors, Histogram of Optical 3D Flow (HOF3D), are extracted from estimated 3D optical flow in the neighborhood of each 4D STIP and made view-invariant. The local HOF3D descriptors are divided using 3D spatial pyramids to capture and improve the discrimination between arm- and leg-based actions. Based on these pyramids of HOF3D descriptors we build a Bag-of-Words (BoW) vocabulary of human actions, which is compressed and classified using Agglomerative Information Bottleneck (AIB) and Support Vector Machines (SVM), respectively. Experiments on the publicly available i3DPost and IXMAS datasets show promising state-of-the-art results and validate the performance and view-invariance of the approach.

The methods presented in previous chapters are totally focused on single camera (2D) based action recognition and we have shown that using model free approach i.e. STIP-local feature based methods are obtaining state-of-the-art results on the action recognition in complex scenes. Human action recognition is not only restricted to the single camera, but it also spans in multi-view camera systems. This chapter deals with the idea of solving action recognition in multi-camera systems.

A 3D data representation is more informative than the analysis of 2D activities carried out in the image plane, which is only a projection of the actual actions. As a result, the projection of the actions will depend on the viewpoint, and not contain full information about the performed activities. To overcome this shortcoming the use of 3D data has been introduced through the use of two or more cameras [35, 60, 167, 200]. In this way the surface structure or a 3D volume of the person can be reconstructed, e.g., by Shape-From-Silhouette (SFS) techniques [173], and thereby a more descriptive representation for action recognition can be established.

2D human action recognition has moved from model-based approaches to model-free approaches using local motion features. In this context, methods based on Spatio-Temporal Interest Points (STIPs) and Bag-of-Words (BoW) are successfully applied to this area. On the contrary, 3D Human action recognition is more confined towards model-based approaches or holistic features. To minimize this gap, we contribute to the field of multi-view human action recognition, by introducing a novel 3D action recognition approach based on detection of 4D Spatio-Temporal Interest Points and local description of 3D motion features extracted from reconstructed 3D data acquired by multi-camera systems.

Opposed to other methods for 3D action recognition, which are solely based on holistic features, e.g. [72, 144, 172, 200], our approach extends the concepts of STIP detection and local feature description for building a Bag-of-Words (BoW) vocabulary of human actions, which has gained popularity in the 2D image domain, to the 3D case. Figure 6.1 shows the schematic diagram of the proposed framework for human action recognition in multi-camera systems.

Rest of the chapter is as follows. Section 6.1 presents a novel 4D (3D space + time) STIP detector. A novel view-invariant 3D motion descriptor called histogram of optical 3D flow (HOF3D) is proposed in Section 6.2. Section 6.3 presents a compact vocabulary representation of the motion features extracted from the 4D STIPs for action representation. Experimantal results on the benchmark multi-camera action recognition datasets are documented in the Section 6.4. Finally, Section 6.5 gives some discussions and conclusion of the proposed framework.

6.1 4D Spatio-Temporal Interest Point Detection

We detect STIPs using the selective STIP detector proposed by [27] (See chapter 5), which first detects spatial interest points (SIPs), then perform surround suppression, impose local spatio-temporal constraints and scale adaption, to obtain a final set of STIPs. Hereafter, we extend the detected STIPs to 4D STIPs using pixel-to-vertex correspondences (Figure 6.2).

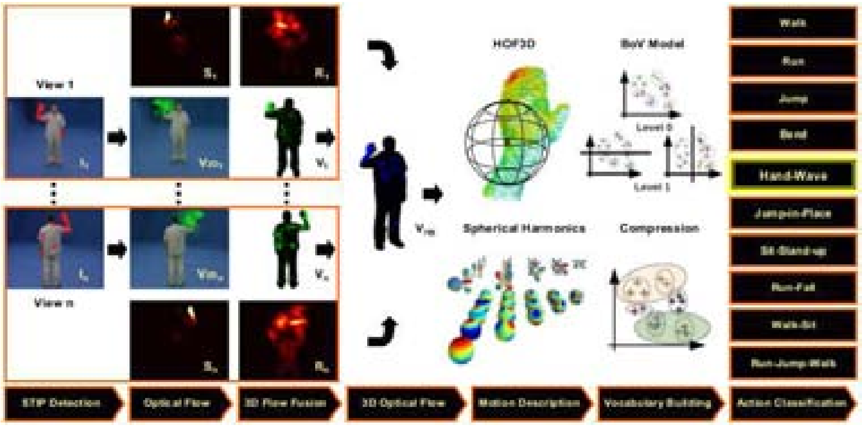


Figure 6.1: A schematic overview of the multi-view human action recognition. Spatio-Temporal Interest Points (STIPs) are detected in multi-view images in a selective manner and Histogram of Optical 3D Flow (HOF3D) is estimated for each view, which is made view-invariant, e.g. by decomposing the representation into a set of spherical harmonic basis functions. Actions are recognized by building a Bag-of-Visual-words (BoV) vocabulary of view-invariant HOF3D descriptors, which is organized in 3D spatial pyramids, and further compressed and classified using Agglomerative Information Bottleneck (AIB) and Class specific Support Vector Machines (SVMs), respectively.

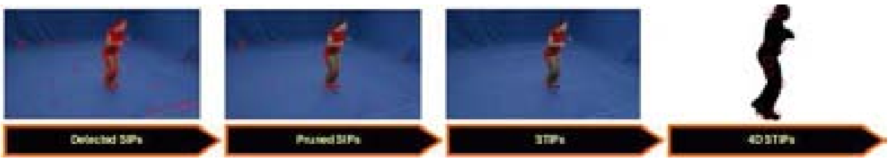


Figure 6.2: Schematic overview of the selective 4D STIPs.

6.1.1 Selective STIPs

The detector applies the basic Harris corner detector [70] and computes the first set of interest points:

$$C_\sigma(x,y) = \frac{I_x^2 I_y^2 - I_{xy}^2}{I_x^2 + I_y^2 + \epsilon} \tag{6.1}$$

where σ is the spatial scale; I_x, I_y and I_{xy} are the partial derivatives over x, y and xy, respectively; and ϵ is a small constant. Apart from the detected SIPs on the human actors, the spatial corners C_σ contain a significant amount of unwanted background SIPs [27].

After detection of STIPs in multi-frame images we extend the resulting interest points into 4D STIPs. For this purpose we use the camera calibration data for the multi-view camera system [60], and project the vertices $\mathbf{p}$ of reconstructed 3D mesh models [173] onto the respective image planes with coordinates (u,v), using the

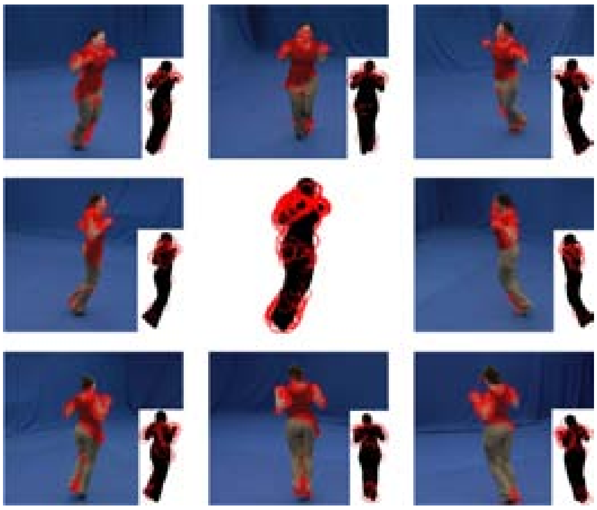


Figure 6.3: Detection of STIPs in multi-frames, and extension to 4D STIPs using 3D reconstructions of the actors and pixel-to-vertex correspondences, for extraction of local 3D motion descriptors.

following set of equations:

$$\begin{aligned}\mathbf{p}_c &= R_i\mathbf{p} + t_i \\ r &= \sqrt{d_x^2 + d_y^2},\ d_x = f_{i,x}\frac{p_{c,x}}{p_{c,z}}, d_y = f_{i,y}\frac{p_{c,y}}{p_{c,z}} \\ (u,v) &= \big(c_{i,x} + d_x(1 + k_{i,1}r), c_{i,y} + d_y(1 + k_{i,1}r)\big)\end{aligned} \tag{6.2}$$

where R and t are the camera rotation matrix and translation vector; f_x and f_y are the x and y components of the focal length f; c_x and c_y are the x and y components of the principal point c, and k_1 is the coefficient of a first order distortion model for the i^{th} camera, respectively. Since multiple vertices might be projected onto the same image pixel, we create a z-buffer containing the depth ordered vertices $\mathbf{p}_d$, and select the vertex with the shortest distance to the respective camera. The distance d is determined with respect to the centre of projection $\mathbf{o}$, as follows:

$$\begin{aligned}\text{z-buffer} &= [\mathbf{p}_{d,1}, \mathbf{p}_{d,2}, \ldots, \mathbf{p}_{d,n}] \\ d &= |\mathbf{p}_i - \mathbf{o}_i|,\ \text{where } \mathbf{o}_i = -R_i^T t_i\end{aligned} \tag{6.3}$$

This has proven to work well for selecting the best corresponding vertices in case of multiple instances [72]. Figure 6.3 present an example of 4D STIP detection.

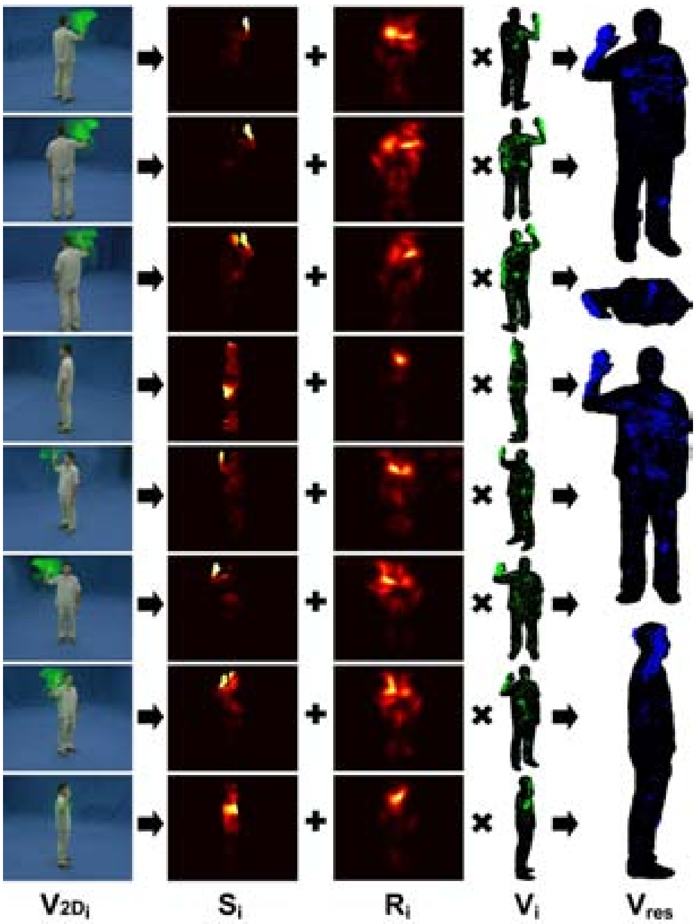


Figure 6.4: A schematic overview of the computation of 3D optical flow $\mathbf{V}_{\mathrm{res}}$, by fusing optical flow estimated in multi-frames $\mathbf{V}_{2D,i}$, extended to 3D flow $\mathbf{V}_i$, and weighted by the significance of local motion $\mathbf{S}_i$ and it reliability $\mathbf{R}_i$.

6.2 Local 3D Motion Description

We detect motion in Multi-frames $\mathcal{F} = (I_1, I_2, \ldots, I_n)$, which is a set of image frames I acquired by n synchronized cameras, using a 3D version of optical flow [72] to produce *velocity annotated point clouds* [178] or *scene flow* [186] (3D optical flow), and combine the estimated 3D optical flow for each view (Fig. 6.4, 6.5 and 6.6). The estimated 3D optical flow is represented efficiently by introducing a local 3D motion descriptor, Histogram of 3D Optical Flow (HOF3D), which is made view-invariant.

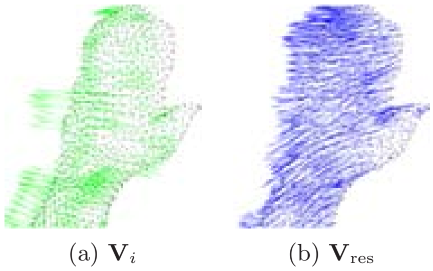


Figure 6.5: Examples of (a) single-view 3D optical flow and (b) combined 3D optical flow.

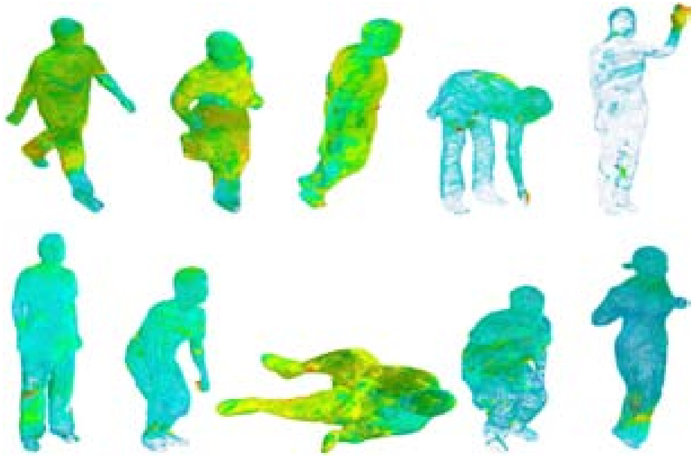


Figure 6.6: Examples of the resulting 3D optical flow for the 10 actions performed by 8 actors in the i3DPost dataset. The velocity vectors are color coded, so blue corresponds to low velocities and yellow/red corresponds to high velocities.

6.2.1 3-Dimensional Optical Flow

Optical flow is computed using the Lucas and Kanade algorithm [119] for each multi-frame $\mathcal{F}_i$ of a multi-view sequence of images $(\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_m)$, and based on data from two consecutive multi-frames $(\mathcal{F}_i, \mathcal{F}_{i-1})$. Each pixel of multi-frame $\mathcal{F}_i$ is annotated with a 2D velocity vector $\mathbf{v}_{2D} = (v_x, v_y)^T$ (see Figure 6.4), resulting in temporal pixel correspondences between multi-frame $\mathcal{F}_i$ and $\mathcal{F}_{i-1}$.

For each pixel in the multi-frames we transform the temporal pixel correspondences into temporal 3D vertex correspondences $(\mathbf{p}_k^i, \mathbf{p}_l^{i-1})$ (Equation 6.2 and 6.3), which can be used to compute 3D velocities $\mathbf{v}_{3D} = (v_x, v_y, v_z)^T = \mathbf{p}_k^i - \mathbf{p}_l^{i-1}$. Figure 6.4 and 6.5.a present examples of estimated 3D optical flow.

The 3D optical flow for each view $\mathbf{V}_i$ is combined into a resulting 3D optical flow $\mathbf{V}_{\text{res}}$, by weighting each component by the significance $\mathbf{S}_i$ of local motion and the reliability $\mathbf{R}_i$ of the estimated optical flow, as given by Equation 6.4:

$$\mathbf{V}_{\text{res}} = \sum_{i=1}^{n} \Big(\alpha \frac{\mathbf{S}_i}{\sum_{k=1}^{n} \mathbf{S}_k} + \beta \frac{\mathbf{R}_i}{\sum_{l=1}^{n} \mathbf{R}_l} \Big) \mathbf{V}_i \tag{6.4}$$

where n is the number of camera views, α and β are weights of the two measurements, such that $\alpha+\beta = 1$ (we set $\alpha = 0.75$ and $\beta = 0.25$). Since we focus on motion vectors, we are interested in robust and significant motion. Therefore, we apply a weight $\mathbf{S} = \sqrt{v_{2D,x}^2 + v_{2D,y}^2}$ to each of the velocity components (v_x, v_y, v_z) falling within the region of interest, determined by the projected silhouettes of the 3D models onto the respective image planes.

In this way we give emphasis to the velocity components based on the total length of the 2D optical flow vector, i.e., the significance of local motions. This had proven to be an important asset, reducing the impact of erroneous 3D motion vectors, when falsified pixel-to-vertex correspondences have been established. The reliability $\mathbf{R}$ is a measure of the "cornerness" of the gradients in the window used to estimate optical flow, and is determined by the smallest eigenvalue $\mathbf{R} = \lambda_2$ of the second moment matrix.

In this way we check for ill conditioned second moment matrices, and give emphasis to flow components based on their reliability. Figure 6.4, 6.5.b and 6.6 show examples of the resulting 3D optical flow.

6.2.2 Histogram of 3D Optical Flow

The extracted 3D motion in the form of $3D$ optical flow is represented efficiently by introducing a local 3D motion descriptor, Histogram of $3D$ Optical Flow (HOF3D), which is based on similar concepts as the HOF image descriptor proposed by Laptev et al. [104]. It is based on a spherical histogram, which is centered in the detected STIP and divided linearly into S azimuthal (east-west) orientation bins and T colatitudinal (north-south) bins (see Figure 6.7). For each bin of the histogram the velocity vector of each vertex falling within that particular bin, within a spherical support region with radius r, is accumulated and weighted by the length of the velocity vector. Hence, the descriptor captures both the location of motion, together with the amount of motion and its direction. We set $S = 8$, $T = 4$ and $r = 100$ mm, resulting in a $S \times T = 32$ dimensional feature vector for each STIP.

In the Scale Invariant Feature Transform (SIFT) [117], partial invariance to the effect of illumination changes on the gradient magnitude is imposed by thresholding and normalizing the feature vector. In the same way we impose partially invariance to the velocity of movements, like in the case where two individuals perform the same action at different speed. Hence, the feature vector gives greater emphasis to the location and orientation, while reducing the influence of large velocity values.

6.2.3 View-Invariance

View-invariance is an essential criterion of feature description and recognition in 3D, since a feature (in our case the direction of extracted motion) might appear very differently depending on the viewpoint. For view-invariant human action recognition it is sufficient to consider the variations around the vertical axis of the human body. In the following we propose four solutions to transform the HOF3D descriptors into view-invariant representations: (i) vertical rotation with respect to the orientation of the normal vector and (ii) the orientation of the velocity vector, (ii) circular bin shifting with respect to the horizontal mode of the histogram, and (iv) by decomposing the representation into a set of spherical harmonic basis functions.

Vertical Rotation

The HOF3D descriptor is rotated around the vertical axis with respect to an azimuthal reference orientation $\angle\theta_{ref}$ of the evaluated STIP: $\angle\theta - \angle\theta_{ref}$. We evaluate two reference orientations. The orientation of the 3D models normal vector (HOF3D$_{\mathrm{norm}}$) and the orientation of the velocity vector of the 3D optical flow (HOF3D$_{\mathrm{flow}}$) at that particular STIP.

Circular Bin Shifting

We perform circular bin shifting of the histogram with respect to the horizontal mode of the histogram (HOF3D$_{\mathrm{mode}}$). The horizontal mode is determined as the set of vertical orientation bins with the largest value. An example is given in Figure 6.8.

Spherical Harmonics

Finally, the HOF3D descriptor is made view-invariant with respect to the vertical axis by decomposing the spherical Histogram representation $f(\theta, \phi)$ into a weighted sum of spherical harmonics (HHOF3D), as given by Equation 6.5.

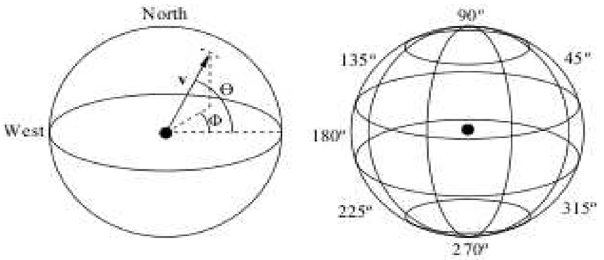


Figure 6.7: The HOF3D descriptor and its subdivision into 8 azimuthal and 4 colatitudinal bins.

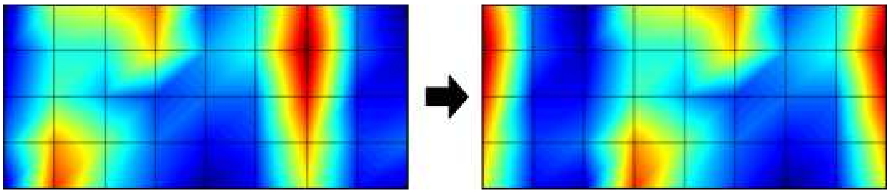


Figure 6.8: Circular bin shifting of the HOF3D histogram with respect to the horizontal mode of the histogram (HOF3D$_{\text{mode}}$).

$$f\left(\theta,\phi\right)=\sum_{l=0}^{\infty}\sum_{m=-l}^{l}A_l^m\,Y_l^m\left(\theta,\phi\right) \tag{6.5}$$

where the term A_l^m is the weighing coefficient of *degree* m and *order* l, while the complex functions $Y_l^m\left(\cdot\right)$ are the actual spherical harmonic functions of *degree* m and *order* l. The complex function $Y_l^m\left(\cdot\right)$ is given by Equation 6.6.

$$Y_l^m(\theta,\phi)=K_l^m\,P_l^{|m|}\left(\cos\theta\right)\,e^{jm\phi} \tag{6.6}$$

The term K_l^m is a normalization constant, while the function $P_l^{|m|}\left(\cdot\right)$ is the *associated Legendre Polynomial*. The key feature to note from Equation 6.6 is the encoding of the azimuthal variable ϕ, which solely inflects the *phase* of the spherical harmonic function and has no effect on the *magnitude*. This effectively means that $||A_l^m||$, i.e. the norm of the decomposition coefficients of Equation 6.5 is invariant to parameterization in the variable ϕ.

The actual determination of the spherical harmonic coefficients is based on an inverse summation as given by Equation 6.7, where N is the number of samples $(S\times T)$, and $4\pi/N$ is the surface area of each sample on the unit sphere.

$$\left(A_l^m\right)_f=\frac{4\pi}{N}\sum_{\phi=0}^{2\pi}\sum_{\theta=0}^{\pi}f\left(\theta,\phi\right)\,Y_l^m\left(\theta,\phi\right) \tag{6.7}$$

In a practical application it is not necessary (or possible, as there are infinitely many) to keep all coefficient A_l^m. Contrary, it is assumed the functions f are band-limited, hence it is only necessary to keep coefficient up to some bandwidth $l=B$, where the dimensionality becomes $D=(B+1)(B+2)/2$. Concretely, we set $B=15$, resulting in 136 coefficients.

6.3 Vocabulary Building and Classification

We apply a BoW model to learn the visual vocabularies of the extracted HOF3D descriptors. We extend the idea of [115] by introducing pyramid levels in the feature space, but instead of applying a pyramid at feature level, as in [113], we apply it at

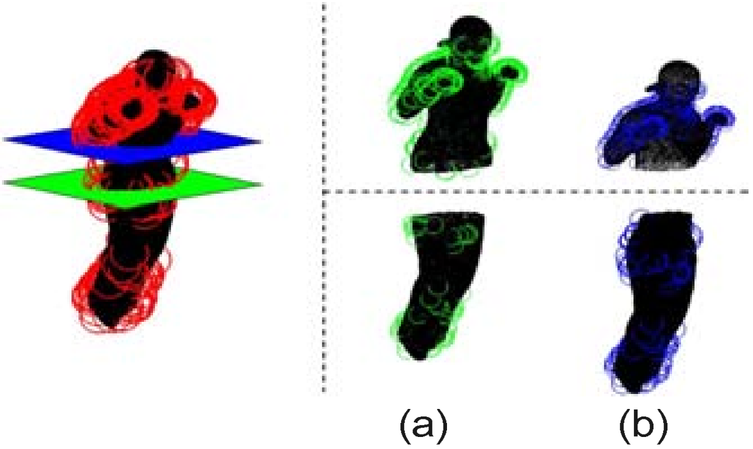


Figure 6.9: 3D spatial pyramid of level 2 with division by a horizontal plane estimated by the center of mass of the reconstructed model (a) and the detected 4D STIPs (b).

STIP level in a 3D coordinate system. This makes the problem of grouping the local features much simpler yet robust, since our STIPs are detected in a selective and robust manner. Finally, we apply vocabulary compression, at each pyramid level, to reduce the dimensionality of the feature space.

6.3.1 3D Spatial Pyramids

Let I_T be the T^{th} frame of the image sequence I and $P^T_{\alpha,\sigma}$ the set of detected STIPs in this frame. We then quantize this the set of detected STIPs into q levels, $\mathcal{S} = \{s_0, s_1, \ldots, s_{q-1}\}$, where the first level s_0 contains all of the detected STIPs. We examine two solutions for pyramid divisions based on a horizontal plane estimated as (i) the center of gravity of the 3D human model ($\mathrm{SP}_{\mathrm{model}}$) and (ii) the center of gravity of the detected STIPs ($\mathrm{SP}_{\mathrm{STIPs}}$). Accordingly, we group the HOF3D descriptors into different levels of the pyramid. The structure of the 2-level 3D spatial pyramid is illustrated in Figure 6.9. This horizontal division helps to capture the distinguishing characteristics of arm- and leg-based actions. We do not apply further pyramid levels or vertical division, since this will conflict with the view-invariance of the approach.

6.3.2 Vocabulary Compression

After dividing the HOF3D descriptors into the described pyramid levels, we create initial vocabularies of a relatively large size (200 words). To reduce the final dimensionality of the feature space, we use vocabulary compression, as in [115], but at each level of the pyramid to achieve a compact yet discriminative visual-word representation of actions.

Let A be a discrete random variable which takes the value of a set of action classes $A = \{a_1, a_2, \ldots, a_n\}$, and W_s be a random variable which range over the set of video-words $W_s = \{w_1, w_2, \ldots, w_m\}$ at pyramid level s. Then the information about A captured by W_s can be expressed by the Mutual Information (MI), $I(A, W_s)$. Now, let $\widehat{W_s} = \{\hat{w_1}, \hat{w_2}, \ldots, \hat{w_k}\}$ for $k < m$, be the compressed video-word cluster of W_s. We can measure the loss of quality of the resulting compressed vocabulary $\widehat{W_s}$, as the loss of MI:

$$Q(\widehat{W_s}) = I(A, W_s) - I(A, \widehat{W_s}) \tag{6.8}$$

To find the optimal compression $\widehat{W_s}$ we use Agglomerative Information Bottleneck (AIB) [170]. We use the described vocabulary compression at each level of the pyramid per class, and obtain a final class-specific compact pyramid representation of video-words.

6.3.3 Action Classification

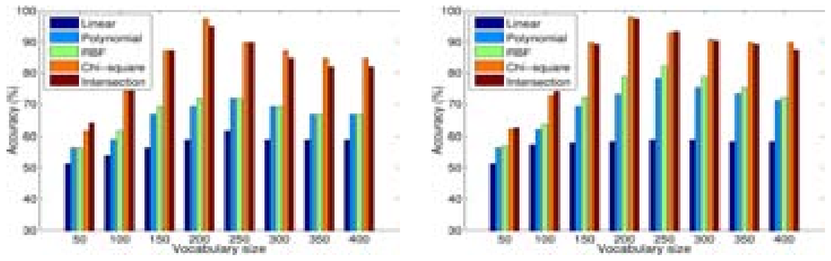
After compression of the video-words at each pyramid level we compute a histograms of the video-words, using the extracted HOF3D descriptors, and concatenate them to a final feature set for SVM learning. We design a class specific χ-square kernel-based SVM, $\mathrm{SVM}_{a_i}(k, h^{a_i}_{W_{a_i}})$ [31]. Where a_i is the i^{th} action class A, k is the SVM kernel and $h^{a_i}_{W_{a_i}}$ is the histogram of action class a_i, computed using the class-specific video-words W_{a_i}. For a test set a_{Test} we detect its action class:

$$i^*_{a_{Test}} = argmax_j \mathrm{SVM}_{a_j}(k, h^{a_{Test}}_{W_{a_j}}), \forall a_j \in A \tag{6.9}$$

We conduct experiments using different SVM kernels and initial vocabulary size, and observe that the χ-square and intersection kernel are the best performing SVM kernels for both the i3DPost and the IXMAS dataset. Hence, we apply the χ-square kernel for all our experiments on human action recognition in Section 6.4. Figure 6.10 shows the average recognition accuracy for the two datasets using a number of different SVM kernels.

6.4 Experimental Results

To test our proposed approach we conduct a number of experiments: (1) action recognition using publicly available multi-view datasets and comparison with the state-of-the-art, (2) an comparison of the different variants of the HOF3D descriptor and 3D spatial pyramids, (3) an incremental analysis of the performance of the vocabulary building process, and (4) evaluation of view-invariance using different camera views for training and testing of the system.



(a) (b)

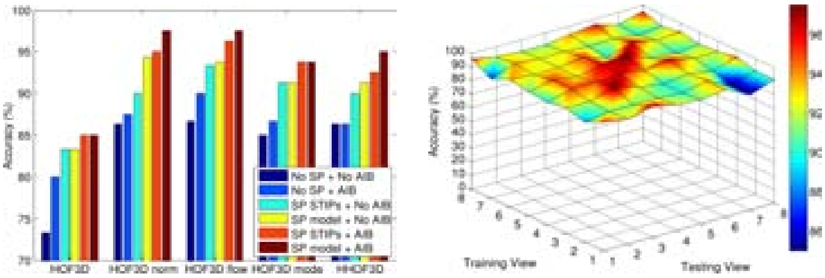
Figure 6.10: Average recognition accuracy for (a) the i3DPost dataset (10 actions) and (b) the IXMAS dataset (13 actions) using *five* different SVM kernels: linear, polynomial, Radial Basis Function (RBF), χ-square and intersection. We have used a polynomial kernel of degree 3.

Table 6.1: State-of-the-art recognition accuracies (%) for the i3DPost dataset. The column named "Dim" states if the methods apply $2D$ image data or $3D$ data. *Gkalelis et al. [61] test on 5 single actions.

Method	Dim	8 actions	10 actions
$\text{HOF3D}_{\text{norm}} + \text{SP}_{\text{model}}$	3D	98.44	97.50
$\text{HOF3D}_{\text{flow}} + \text{SP}_{\text{model}}$	3D	96.88	97.50
$\text{HOF3D}_{\text{mode}} + \text{SP}_{\text{model}}$	3D	95.31	93.75
$\text{HHOF3D} + \text{SP}_{\text{model}}$	3D	93.75	95.00
$\text{HOF3D}_{\text{norm}} + \text{SP}_{\text{STIPs}}$	3D	96.88	95.00
$\text{HOF3D}_{\text{flow}} + \text{SP}_{\text{STIPs}}$	3D	98.44	96.25
$\text{HOF3D}_{\text{mode}} + \text{SP}_{\text{STIPs}}$	3D	93.75	93.75
$\text{HHOF3D} + \text{SP}_{\text{STIPs}}$	3D	93.75	92.50
Holte et al. [72]	3D	92.19	78.75
Iosifidis et al. [79]	2D	90.88	-
Gkalelis et al. [61]	2D	90.00*	-

6.4.1 Evaluation on i3DPost

For the first test we use the data available for all 8 camera views and the full action set of 10 actions (single and combined). Additionally, we split the combined action up into two additional single actions [79], resulting in a total of 8 single actions. We perform leave-one-out cross validation, hence, we use one actor for testing, while the system is trained using the rest of the dataset. Table 6.1 presents the results of our approach using the described variants of the HOF3D descriptors and 3D spatial pyramids in comparison to Iosifidis et al. [79] and Gkalelis et al. [61]. The results show comparable performance for the descriptor and pyramid variants, but with a slightly better overall performance using $\text{HOF3D}_{\text{norm}} + \text{SP}_{\text{model}}$, followed up by $\text{HOF3D}_{\text{flow}} + \text{SP}_{\text{model}}$ and $\text{HOF3D}_{\text{flow}} + \text{SP}_{\text{STIPs}}$. For the 8 single actions, the accuracy of $\text{HOF3D}_{\text{norm}} + \text{SP}_{\text{model}}$ and $\text{HOF3D}_{\text{flow}} + \text{SP}_{\text{STIPs}}$ are **98.44**%, while for the full action set of 10 actions, the accuracy of $\text{HOF3D}_{\text{norm}} + \text{SP}_{\text{model}}$ and $\text{HOF3D}_{\text{flow}}$ +



(a) (b)

Figure 6.11: (a) Plot of the recognition accuracy of the four HOF3D variants with and without spatial pyramids or AIB compression, and (b) plot of the recognition accuracy as a function of the applied camera views for training and testing (i3DPost).

$\mathrm{SP}_{\mathrm{model}}$ are **97.50%**. The other two descriptor variants, $\mathrm{HOF3D}_{\mathrm{mode}}$ and HHOF3D, have slightly lower but comparable performance. These results are consistent with our expectations, since HHOF3D is an approximation of HOF3D by decomposing the representation into spherical harmonic basis functions within a certain bandwidth, while the circular bin shifting variant $\mathrm{HOF3D}_{\mathrm{mode}}$ can be seen as a fast but more coarse vertical rotation. In general the 3D spatial pyramid divisions based on a horizontal plane estimated as the center of gravity of the 3D human model ($\mathrm{SP}_{\mathrm{model}}$) performs slightly better considering all descriptors variants. This might be due to better location and precision of the horizontal plane, compared to the one estimated as the center of gravity of the detected STIPs ($\mathrm{SP}_{\mathrm{STIPs}}$), which can variate due to the amount of detected STIPs.

Incremental Analysis

Next we conduct an incremental analysis to investigate the performance boost by applying the 3D spatial pyramids and vocabulary compression. Figure 6.11.a shows the recognition accuracy for the *five* HOF3D variants (including the raw non-view-invariant HOF3D descriptor) with and without $3D$ spatial pyramids ($\mathrm{SP}_{\mathrm{model}}$ and $\mathrm{SP}_{\mathrm{STIPs}}$) or AIB vocabulary compression. The plot clearly indicates the performance boost by using spatial pyramids and compression for all descriptor variants. The largest performance increase occurs when applying spatial pyramids (∼5.5%). The vocabulary compression improves the average accuracy by ∼1.5%, however, when AIB is applied at pyramid level the performance boost is more significant (∼3%). Note that the recognition accuracy is significant lower for the raw non-viewinvariant HOF3D descriptor. Although the BoV model is able to handle this to some extent, this clearly shows the importance of making the descriptor view-invariant.

View-Invariance

To observe the view-invariance of our approach we evaluate its capability to recognize actions using different camera views for training and testing. We train and test the system by detecting STIPs, extracting $\text{HOF3D}_{\text{norm}} + \text{SP}_{\text{model}}$ descriptors and building vocabularies for classification for each of the 8 views, separately. Figure 6.11.b shows a plot of the results, when recognizing all 10 actions using each combination of the 8 views for training and testing. As can be seen from the plot, the recognition accuracy is quite stable over all view combinations ($\sim$91% ± 6%). Note that only a small increase in the accuracy can be observed, when training and testing with the same view. This is shown by a bit higher recall rates for the diagonal elements in Figure 6.11.b, except for view combination 2-2 and 7-7, where some actions like *waving*, *walking* and *jump* etc. are more difficult to discriminate between from these viewpoints for multiple video sequences. For example, the waving hand is in some cases more or less hidden behind the head.

6.4.2 Evaluation on IXMAS

Table 6.2 presents the results of our approach using the HOF3D descriptors and 3D spatial pyramids (SP) in comparison to the state-of-the-art methods. Some authors only test on 11 actions performed by 10 actors (the test setup proposed by Weinland et al. [200]), while others evaluate their algorithms on the full dataset. Hence, to compare our approach to other works, we apply both test setups. As shown in the table our approach achieves a perfect recognition for both the 11 and 13 action setup, and thereby outperforms other proposed methods. The recognition accuracies are identical for all HOF3D descriptor and pyramid variants. Futhermore, this validates that our approach can be used for multi-view data of lower data quality and resolution.

6.4.3 Cross-data Evaluation

To show the generality of our approach, we have performed a cross-data evaluation test. We select three common actions between IXMAS and i3DPost: *walk*, *wave* and *sit*. Using this setup, we achieve a recognition rate of 88.89% when training on i3DPost and test on IXMAS, and 87.50% when training on IXMAS and testing on i3DPost. These accuracies are somewhat lower than the evaluation based on one single dataset for both training and testing, however, the results validate the generality of our approach.

6.5 Conclusion

We have presented a 4D STIP and local 3D motion descriptor-based approach for human action recognition using 3D data acquired by multi-camera setups. We contribute to this field by: (1) the design of a 4D STIP detector, which operates in a

Table 6.2: State-of-the-art recognition accuracies (%) for the IXMAS dataset. The column named "Dim" states if the methods apply $2D$ image data or $3D$ data.

Method	Dim	11 actions	13 actions
HOF3D + SP	3D	100.00	100.00
Turaga et al. [181]	3D	98.78	-
Weinland et al. [200]	3D	93.33	-
Pehlivan et al. [144]	3D	90.91	88.63
Vitaladevuni et al. [188]	2D	87.00	-
Haq et al. [68]	2D	83.69	-
Weinland et al. [199]	2D	83.50	-
Liu et al. [113]	2D	-	82.80
Liu et al. [114]	2D	82.80	-
Weinland et al. [198]	2D	81.27	-
Lv et al. [121]	2D	-	80.60
Tran et al. [180]	2D	-	80.22
Cherla et al. [34]	2D	-	80.05
Liu et al. [111]	2D	-	78.50
Yan et al. [207]	3D	78.00	-
Junejo et al. [87]	2D	74.60	-
Junejo et al. [86]	2D	72.70	-
Reddy et al. [152]	2D	-	72.60
Farhadi et al. [49]	2D	58.10	-

selective manner by incorporating surround sup- pression and local spatio-temporal constraints. (2) Introducing a novel local 3D motion descriptor (HOF3D) for description of estimated 3D optical flow, and examine a number of solutions to make it view-invariant. (3) Based on 3D spatial pyramids of HOF3D descriptors we build a BoW vocabulary of human actions, which is compressed and classified using AIB and SVM, respectively. (4) We have reported superior performance on the publicly available i3DPost and IXMAS datasets, investigated the incremental performance boost of the proposed 3D spatial pyramids and vocabulary compression, and evaluated the view-invariance of the approach. To the best of our knowledge, we are the first to detect spatio-temporal interest points in 4D to extract local 3D feature descriptors in a multi-view framework.

In future work it would be interesting to adapt the method to single view depth sensors (Time-of-Flight range cameras and the Kinect sensor [164]), which in general are more flexible and applicable. Multi-camera systems are limited to a specific area of interest, due to its nature. However, it also helps to uncover occluded action regions from different views in the global 3D data, and allows for extraction of informative features in a more rich 3D space, than the one captured from a single view.

Chapter 7

Large scale continuous visual event recognition

"We can understand almost anything, but we can't understand how we understand."

by Albert Einstein.

In this chapter we propose a novel method for continuous visual event recognition (CVER) on a large scale video dataset using max-margin Hough transform framework. Due to high scalability, diverse real environmental state and wide scene variability, direct application of local "region of interest" detection, such as spatio-temporal interest point (STIP), on the whole dataset is practically infeasible. To address this problem we apply a motion region extraction technique which is based on motion segmentation and region clustering to identify possible candidate 'event of interest' as a preprocessing step. On these candidate regions a STIP detector is applied and local motion features are computed. For the activity representation we use generalized Hough transform framework where each feature point casts a weighted vote for the possible activity class center. A max-margin frame work is applied to learn the feature codebook weight. For activity detection, peaks in the Hough voting space are taken into account and initial event hypothesis is generated using the spatio-temporal information of the participating STIPs. For event recognition a verification Support Vector Machine is used. An extensive evaluation on benchmark large scale video surveillance dataset (VIRAT) and as well on a small scale benchmark dataset (MSR) shows that the proposed method is applicable on a wide range of continuous visual event recognition applications having extremely challenging conditions.

Recently, research interest is moving towards *continuous visual event recognition*

(CVER) where the goal is to both recognize an event and to localize the corresponding space-time volume from large continuous video [138]. This area is more closely related to the real world video surveillance analytics need than the current research which aims to classify a prerecorded video clip of a single event. Accurate CVER would have direct and far reaching impact in surveillance, video-guided human behaviour analysis, assistive technology and video archive analysis.

The task of CVER, i.e. the activity recognition on large scale real world video surveillance dataset, is an extremely challenging task and current state-of-the methods for 2D small scale action recognition (See Chapter 2) become infeasible to apply. One of the main challenges for CVER is the scalability, e.g. a typical CVER dataset contains 23 event types distributed throughout 29 hours of video. The other difficulties are due to i) natural appearance since the events are recorded in a real world scenario, ii) huge spatial and temporal coverage which affects the video resolution, e.g. the human heights within videos range 25 $\sim$ 200 pixels constituting 2.4 $\sim$ 20% of the heights of the recorded videos with an average being about 7%, iii) diverse event types and iv) huge variability in view-points, scenes and subjects (See Figure 7.1). Our method for event detection is related to several ideas recurring in the literature.

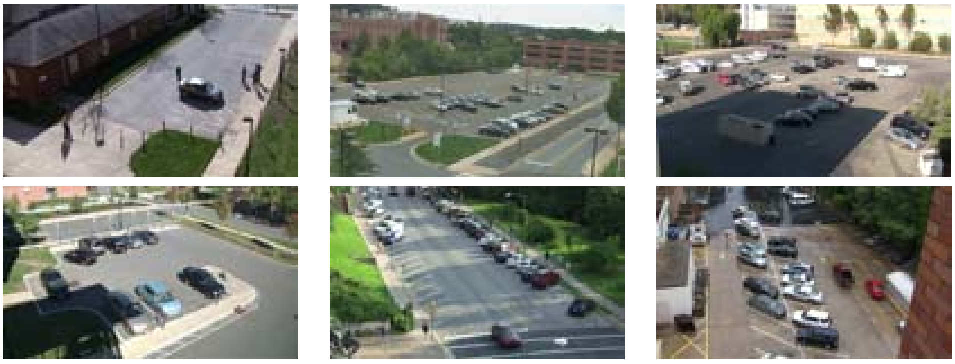


Figure 7.1: Examples of 6 scenes present in the VIRAT dataset for large scale activity detection which explains different challenges: realism and natural scenes, diversity, quantity and wide range of scene resolution.

Firstly, we use STIP detector which is successfully applied in 2D action recognition problems [27, 159, 100, 44, 115]. Several local features are computed such as histogram of oriented optical flow (HOF) [32], histogram of oriented gradient in 3D (HOG3D) [23] and extended SURF (ESURF)[202] at the detected STIPs. We use the idea of *local appearance codebook*[168] including bag-of-word approach [134, 38] to group the detected features into a set of *visual words* that represent an event class.

The second idea is to use the generalized Hough transformation for object detection into event detection in videos. Originally developed for detecting straight line [45], Hough transforms are generalized to use for detecting generic parametric shapes [9]. Recently, the concept of generalized Hough transform is successfully used for detecting object class instances tracking and action recognition [106, 107, 53, 123, 140, 141]. The concept of generalized Hough transform usually refers to any detection process based on an additive aggregation of evidence, Hough votes, coming from local image/video elements. Such aggregation is performed in a parametric space, Hough space, where each point corresponds to the existence of an instance in a particular configuration. The Hough space may be a product set of different locations, scales and aspects etc. The detection process is then reduced to finding maxima peaks in the sum of all Hough votes in the Hough space domain, where the location of each peak gives the configuration of a particular detected object/event instance.

The implicit shape model of Leibe et al. [106] and the max-margin hough transformation of Maji et al. [123] serve a baseline for our work. These works mainly focus on object detection. During training, they augment each visual words in the codebook with the spatial distribution of the displacements between the object center and the respective visual word location. Using max-margin setup the weights of each visual words are learned. At the detection time, these spatial distributions are converted into Hough votes withing the Hough transformation framework. The weights of the visual words are also used for extra information to the Hough votes.

To incorporate this idea into CVER, we need to extend the dimensionality of the voting space since now each STIP will vote for a parallelepiped center i.e. the event center. To make it easier to to understand, we scale each candidate event into a normalized cube and during training the interest point (feature) distributions along the cube center is learned for each event class. The scale information is also saved so that by using simple reverse conversion the normalized cube can be transformed into the actual event parallelepiped. After obtaining a set of visual words from detected event features, a max-margin frame work similar to [123] is applied for learning weights of each visual words for each event class. For a test candidate region, the detected interest points (features) are matched with the event class visual words and weighted votes for the possible event center are obtained in the Hough voting space. The votes corresponding to the peaks of Hough space reveal the possible hypothesis of the detected events in the actual video. Finally, a verification Support Vector Machine (SVM) designed for the particular event class is used to obtain the recognition score. Figure 7.2 shows the schamatic representation of the overall activity detection framework. To test our approach we use the large scale CVER dataset, VIRAT, proposed by [138]. Our result shows the state-of-the-art performance. To show the diversity of our method we choose small scale video search dataset MSR [217] and also obtained above state-of-the-art result.

Rest of the chapter is as follows: Section 7.1 presents region clustering based motion segmentation technique to identify the candidate region of interest. Max-margin generalized Hough transofrmation framework for activity detection is described in the Section 7.2. We present our experimental results in the Section 7.3.Finally, the conclusions are given in the Section 7.4.

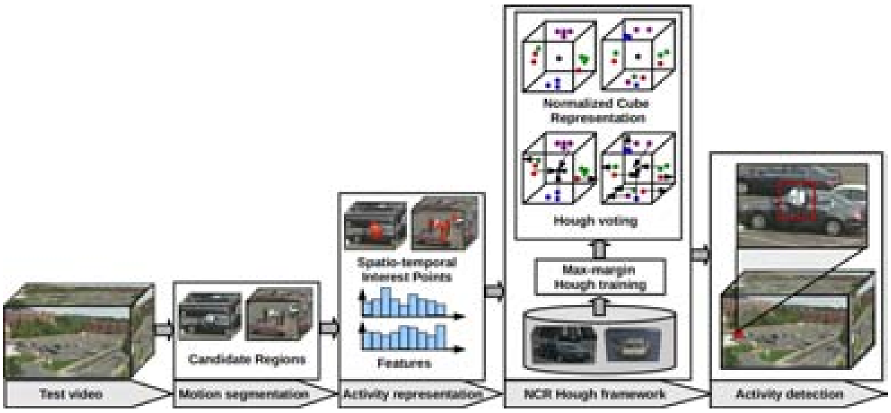


Figure 7.2: Schematic overview of the activity detection framework proposed in this chapter.

7.1 Region clustering based motion segmentation

To tackle the scalability issue of CVER it is important to reduce the search space. We apply a motion segmentation technique to identify roughly the motion regions where *the event of interest* may appear.

7.1.1 Background subtraction

As the first step of motion segmentation we apply a background segmentation technique as in [174]. In this method, each image pixel is modeled as a mixture of Gaussians and use an on-line approximation to update the model. The Gaussian distributions of the adaptive mixture model are then evaluated to determine which are most likely to result from a background process. Each pixel is classified based on whether the Gaussian distribution, which represents it most effectively, is considered part of the background model. We apply the OpenCV in-built functions for background segmentation (See Algorithm 8). After obtaining the result of background subtraction we apply the connected component algorithm [156] to obtain motion regions per frame.

7.1.2 Region clustering and candidate event identification

Motion regions obtained by using the above step contain a large number of broken parts. To join those broken parts a region clustering method is applied as described in Algorithm 9. In this approach, we first sort the obtained initial broken regions based on their Y-axis coordinate. Different broken parts are joined if they are within a horizontal and vertical thresholds, τ_x, τ_y, respectively. These thresholds are average bounding box dimensions of the human/(human, object) participating in the event of interest, which can be obtained from the training. We use the values $(\tau_x, \tau_y) = (80, 70)$ in our experiments.

After the region joining, a candidate region extraction as described in Algorithm 10, 11 is applied. Candidate region extraction is based on the *action heuristics*. The event of interest in VIRAT and MSR datasets are not moving in consecutive frames, since the actions are of type, “getting inside car”, “open the trunk of a car” and “loading objects in the car” etc. or “clapping”, “waving” and “boxing”. So if these events are occurring along with other moving actions like “walking” and “running” we are guaranteed to obtain fixed regions, corresponding to events of interest, along with some moving regions , corresponding to the moving actions, in consecutive frames. Based on these heuristic identifying motion region is simply to identify region having some permissible region overlap in consecutive frames.

To realize this goal, we apply a region search technique to first obtain a chain of motion region that have a permissible region overlap within a threshold, τ_d. This chain of motion region may contain some false alarm like, a person walking at a slow rate. To avoid such outliers, we apply a second method (Algorithm 11) which takes only the regions having higher overlapping (τ_a) in consecutive frames. As a final step, we apply a region merging by putting a considerable frame gap (τ_l). With this, two candidate regions having τ_l frame gap and overlapping area withing τ_a are merged together. We use $\tau_a = 45\%$ and $\tau_l = 5$ in all our experiments.

Algorithm 8 Background segmentation using mixture of Gaussian model

```
Require: imStack: Having $N$ frames of size $(H \times W)$.
Ensure: Moation regions (foreground) per frame.
Initialize Gaussian parameters, $gP$.
for $i \in N$ do
    if $(i == 1)$ then
        $bgModel = cvCreateGaussianBGModel(imStack(:, :, i), gP)$.
    else
        $cvUpdateBGStatModel(imStack(:, :, i), bgModel)$.
        $findComonentLabel(bgModel.foreground, regionInfo, noComponent)$.
        $save(regionInfo, noComponent)$.
    end if
end for
$cvReleaseBGStatModel(bgModel)$.
```

7.2 Max-margin Hough transform framework for event detection

The general idea to apply a Hough transformation framework [106] into an action detection problem is to compute the probabilistic score which is obtained by adding

Algorithm 9 Region clustering algorithm

```
Require: regionInfo, noRegion: Region information per image frame.
Ensure: Pruned motion region of each image frame.
$regionInfo = sort(regionInfoY)$.
repeat
    $clusterNo = 0$.
    for $regCurr \in regionInfo$ do
        $clusterNo = clusterNo + 1$.
        $[x_C, y_C] = getCenter(regCurr)$.
        $regCurr.cluster = clusterNo$.
        for $regNext \in (regionInfo \setminus regCurr)$ do
            $[x_{C_N}, y_{C_N}] = getCenter(regNext)$.
            $x_D = fabs(x_C - x_{C_N})$.
            $y_D = fabs(y_C - y_{C_N})$.
            if $(x_D \leq \tau_x) \wedge (y_D \leq \tau_y) \wedge (\sim regNext.cluster)$ then
                $regNext.cluster = regCurr.cluster$.
            end if
        end for
    end for
    Initialize $flag$.
    for $clusterInd \in clusterNo$ do
        for $reg \in regionInfo$ do
            if $\sim flag(clusterInd)$ then
                $[x_L, y_L, x_R, y_R] = getCoordinate(reg)$
                $flag(clusterInd) = 1$
            else
                $[x_L, y_L] = min([x_L, y_L], getCoordinateL(reg))$
                $[x_R, y_R] = max([x_R, y_R], getCoordinateR(reg))$
            end if
        end for
        $clusterInfo.clusterNo = clusterInd$.
        $clusterInfo = putInfo([x_L, y_L, x_R, y_R])$;
    end for
    $regInfo = clusterInfo$.
    if $clusterNo == noRegion$ then
        $CONVERGENCE = TRUE$.
    end if
    $noRegion = clusterNo$.
until $CONVERGENCE \vee MAXITER$
```

up the votes from D-dimensional feature vectors extracted from a candidate video event in a Hough space $\mathcal{H} \subseteq \mathbb{R}^H$. In our case we apply a spatio-temporal interest point (STIP) detector [27] on candidate event (Figure 7.3) and the feature vector is the concatenation of HOOF, HOG3D and ESurf.

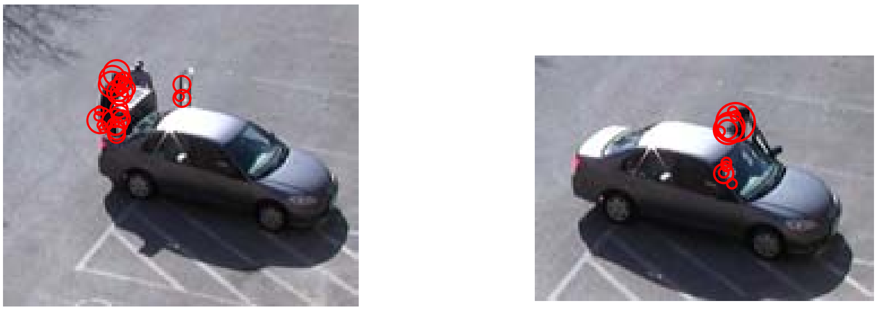
Algorithm 10 Find candidate event/video of interest

```
Require: $pRegInfo$, $noPReg$: Pruned region information per image frame.
Ensure: Candidate event/video of interest.
for $imC \in N$ do
    for $reg \in pRegInfo_{imC} \wedge notChecked(reg)$ do
        $checked(reg)$.
        $[x_C, y_C] = getCenter(pRegInfo_{imC})$.
        $putInfo(regTack, getInfo(pRegInfo_{imC}))$.
        for $imNC \in (N \setminus imC)$ do
            $flag = 0$.
            for $regN \in pRegInfo_{imNC} \wedge notChecked(regN)$ do
                $[x_{C_N}, y_{C_N}] = getCenter(pRegInfo_{imNC})$.
                if $dist([x_C, y_C], [x_{C_N}, y_{C_N}]) \leq \tau_d$ then
                    $checked(reg)$.
                    $flag = 1$
                    $putInfo(regTack, getInfo(pRegInfo_{imNC}))$.
                end if
            end for
            if $flag == 0$ then
                $extractCandidates(candidate, regTrack)$.
                $break$.
            end if
        end for
        if $flag == 1$ then
            $extractCandidates(candidate, regTrack)$.
        end if
    end for
end for
for $reg \in candidate$ do
    for $regN \in (candidate \setminus reg)$ do
        if $overlap(reg, regN)$ then
            if $(abs(reg.End) - regN.Start) \leq \tau_l)$ then
                $candidate = merge(reg, regN)$.
            end if
        end if
    end for
end for
$saveCandidate(candidate)$
```

So formally, let $\mathcal{A}$ be a candidate event having center at $r = \{x, y, t, s\}$ where $\{x, y, t\}$ is the coordinate of the center and s is the scale of the detection. The feature vector f_i is computed at a location l_i. l_i is basically associated with a STIP $\{x_i, yi, t_i, s_i\}$. So, the probabilistic score, $S(\mathcal{A}, r)$, can be obtained following [106, 123]

Algorithm 11 Extract candidate event/video of interest

```
Require: $regTrack$: Information of motion region in consecutive frames having some permissible overlap.
Ensure: Candidate event/video of interest.
for $reg \in regTrakc$ do
    $regNext = getNext(regTrack, reg)$.
    if $overlap(reg, regNext) \leq \tau_a$ then
        $merge(reg, regNext)$.
    end if
    $candidate = getInfo(reg)$.
end for
```



(a) (b)

Figure 7.3: Detected STIPs on the two events (a) "loading" and (b) "getting into vehicle" activity of the VIRAT dataset..

$$\begin{aligned} S(\mathcal{A}, r) &= \sum_j p(\mathcal{A}, r, f_j, l_j) && (7.1) \\ &= \sum_j p(f_j, l_j) p(\mathcal{A}, r | f_j, l_j) && (7.2) \end{aligned}$$

Let C_i denotes the i^{th} codebook entry of the vector quantized space of features f. Assuming a uniform prior over features, f_j, local patches, l_j together with codebook entries, C_i, the score we get:

$$\begin{aligned} S(\mathcal{A}, r) &= \sum_j p(\mathcal{A}, r | f_j, l_j) && (7.3) \\ &= \sum_{i,j} p(C_i | f_j, l_j) p(\mathcal{A}, r | C_i, f_j, l_j) && (7.4) \end{aligned}$$

This equation can further be simplified using the argument that $p(C_i|f_j, l_j)$ is equivalent to $p(C_i|f_j)$ since the codebook entries, C_i, are based only on the features, f_j. Furthermore, the term $p(\mathcal{A}, r|C_i, f_j, l_j)$ depends only on the matched codebook C_i and l_j,

$$\begin{aligned} S(\mathcal{A}, r) &\propto \sum_{i,j} p(C_i|f_j)p(\mathcal{A}, r|C_i, l_j) && (7.5) \\ &= \sum_{i,j} p(C_i|f_j)p(r|\mathcal{A}, C_i, l_j)p(\mathcal{A}|C_i, l_j) && (7.6) \end{aligned}$$

The first term, $p(C_i|f_j)$ is the likelihood that the feature f_i is associated with the codebook entry C_i. This we define as,

$$p(C_i|f) = \begin{cases} \frac{1}{Z}exp(-\gamma sim(C_i, f)) & \text{if } sim(C_i, f) \leq t \\ 0 & \text{otherwise} \end{cases} \tag{7.7}$$

where Z is the constant of the probability distribution $p(C_i|f)$ and $sim(C_i, f) = \frac{1}{d(C_i,f)}$ where $d(C_i, f)$ is the distance between the feature f and the codeword C_i. γ and t are positive constant.

The second term, $p(r|\mathcal{A}, C_i, l_j)$ is the probabilistic Hough vote for the activity center r which is estimated in the training phase by observing the distribution of the location of the codebooks relative to the activity center. The third term, $p(\mathcal{A}|C_i, l_j)$ is the weight of the codebook entry emphasizing the confidence of the codebook C_i at location l_j that matches the activity $\mathcal{A}$. Final term, $p(\mathcal{A}|C_i, l_j)$, can further be simplified by assuming that the probability $p(\mathcal{A}|C_i, l)$ is independent of the location,

$$p(\mathcal{A}|C_i, l) = p(\mathcal{A}|C_i) \propto \frac{p(C_i|\mathcal{A})}{p(C_i)} \tag{7.8}$$

While applying this framework in CVER (large scale activity detection) the computation of the second term, $p(r|\mathcal{A}, C_i, l_j)$, becomes complicated due to the in-feasibility to apply any state-of-the-art STIP detector or feature extraction method directly on the large scale video. We must work on the possible candidate regions either obtained from the ground truth or by applying a motion segmentation method (see Section 7.1) to extract motion regions where the candidate event of interest may appear. Then $p(r|\mathcal{A}, C_i, l_j)$ is the collection of distances $\{d_{x_j}, d_{y_j}, d_{t_j}, d_{s_j}\}$ between the STIP $\{x_j, yj, t_j, s_j\}$ associated to l_j and the activity parallelepiped center r.

7.2.1 Learning the codebook weight using max-margin framework

The Equation 7.6 can further be simplified as a weighted vote for event video location over all codebook entries C_i. The key idea, as described in [123], is to observe that the score $S(\mathcal{A}, r)$ is a linear function of $p(\mathcal{A}|C_i)$ (Equation 7.8). Using this idea Equation 7.6 can be expressed as,

$$\begin{aligned}
S(\mathcal{A}, r) &\propto \sum_{i,j} p(r|C_i, l_j) p(C_i|f_j) p(\mathcal{A}|C_i, l_j) && (7.9)\\
&= \sum_{i,j} p(r|C_i, l_j) p(C_i|f_j) p(\mathcal{A}|C_i) && (7.10)\\
&= \sum_{i} p(\mathcal{A}|C_i) \sum_{j} p(r|C_i, l_j) p(C_i|f_j) && (7.11)\\
&= \sum_{i} \lambda_i \times q_i(r) = \lambda^T Q(r) && (7.12)
\end{aligned}$$

where $Q^T = [q_1 q_2 \dots q_k]$ is the activation vector and q_i is given by,

$$q_i = \sum_{j} p(r|C_i, l_j) p(C_i|f_j) \tag{7.13}$$

For a given event and identity the summation over j is constant and is only a function of the observed features, locations (STIPs) and the estimated distribution over the centers for the codebook entry C_i.

To learn the weight vector λ, a max-margin optimization approach as describe in [123] is used. Starting from a set of training examples, $\{(q_i, y_i)\}_{i=1}^{L}$, where $y_i \in \{+1, -1\}$ is the label and q_i is the i^{th} training activity, we compute the activations $Q_i = Q(q_i)$ for each example by adding up the votes for each feature f_j extracted at location STIP l_j according to the Equation 7.13. So, the score assigned by the model to the instance i is $\lambda^T Q_i$. Weights are learned by maximizing this score on correct classification of events over the incorrect ones. This is done using a max-margin frame work ([123]),

$$\begin{aligned}
min_{\lambda,b,\xi} \quad & \frac{1}{2}\lambda^T\lambda + K\sum_{i=1}^{M} \xi_i && (7.14)\\
s.t.: \quad & y_i(\lambda^T Q_i + b) \geq (1 - \xi_i) && (7.15)\\
& \lambda \geq 0, \xi_i \geq 0, \forall i = 1, 2, \dots L && (7.16)
\end{aligned}$$

This optimization is similar to the optimization problem of a linear Support Vector Macine [37], with an additional positive constrain on the weights. We use a traditional optimization package, CVX [1] [64], for solving this problem.

[1] http://standford.edu/∼boyd/cvx

7.2.2 Overall detection technique

The proposed max-margin frame work is run on the each extracted candidate region by the region extraction algorithm proposed in Section 7.1. After obtain the votes in our Hough space $\mathcal{H} \subseteq \mathbb{R}^4$ a mean shift based clustering algorithm [36] is used to identify the location of the peaks. After getting the peaks initial hypothesis of the event in actual video coordinate is obtained. This is computed based on the information of the participating STIPs in the maxima of the $\mathcal{H}$.

From this initial hypothesis the candidate region is extracted and fed a verification SVM. This verification SVM is similar to the action recognition SVM approach in [27]. This is learned using the training activity features. In this paper, we use intersection kernel and no feature pyramid is used. The score obtained from the verification SVM is used as a final score of the detected region. This score is later used to compute ROC curve and average precision (AP).

7.3 Experimental results

To validate our proposed approach, experiments on two benchmark datasets are performed. We use VIRAT dataset [138] for large scale event detection. Since not much work has been done on this dataset, Microsoft Research Action (MSR) Dataset II, [25, 217] which is a small scale action detection dataset, is used to compare our approach with other state-of-the-art methods.

7.3.1 Activity recognition on small scale

To perform experiments on small scale dataset we use MSR action dataset II. Most state-of-the-art approaches like, [215, 25, 217] use this dataset as cross-data action recognition where KTH [2] is used as training dataset and MSR is used as test dataset. All these methods apply *model adaptation* to perform the cross-dataset action detection. Since our approach does not design to perform cross-dataset action detection rather our goal is to present generalized Hough transformation framework on action detection, we split MSR action dataset II into two groups: first 16 videos as *training set* and rest 38 videos as *test set*. Ground truth annotations are used to separate actions in the *training set*. We first apply our motion segmentation approach described in Section 7.1 and apply recall test using ground truth annotation to validate the proposed region extraction method. We obtain 100% recall in all three actions, *hand waving*, *hand clapping* and *boxing*, present in *test set*.

To evaluate the the detection results of our algorithm, we follow the same technique as proposed by Cao et al. [25]. Let $\mathbf{Q}^g$ be the ground truth instances, $\mathbf{Q}^g = \{Q_1^g, Q_2^g, \dots, Q_m^g\}$, and $\mathbf{Q}^g$ be the instances detected by the algorithm, $\mathbf{Q}^d = \{Q_1^d, Q_2^d, \dots, Q_m^d\}$. $H(Q_i^g)$ denotes whether a ground truth instance Q_i^g is detected

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

Table 7.1: Comparison of average precision (AP) values of the 3 actions of MSR dataset with other state-of-the-art approaches. Note that both Yu et al. [215] and Cao et al. [25] use cross-data action detection approach with full set of MSR, instead we train on first 16 videos of MSR and test on the rest 38 videos to comply with our proposed algorithm setup.

AP	Boxing	Hand clapping	Hand waving
Our approach	**0.4571**	0.2327	**0.4938**
Yu et al. [215]	0.3029	**0.3155**	0.4923
Cao et al. [25]	0.1748	0.1316	0.3671

and $T(Q_j^d)$ denotes if a detected instance Q_j^d is properly matched with the ground truth set $\mathbf{Q}^g$. These values can be calculated as,

$$H(Q_i^g) = \begin{cases} 1 & \text{if } \exists Q_k^d, \text{ s.t. } \frac{|Q_k^d \cap Q_i^g|}{|Q_i^g|} > \delta_1 \\ 0 & \text{otherwise} \end{cases} \tag{7.17}$$

$$T(Q_j^d) = \begin{cases} 1 & \text{if } \exists Q_k^g, \text{ s.t. } \frac{|Q_k^g \cap Q_j^d|}{|Q_j^d|} > \delta_2 \\ 0 & \text{otherwise} \end{cases} \tag{7.18}$$

where $|\cdot|$ denotes the area of the video instance and δ_1, δ_2 use to judge the overlapping ration. δ_1 and δ_2 are set to 0.125 as proposed by Cao et al. [25].

Given a set of detected instances the *precision* and *recall* can be computed using the values of H and T,

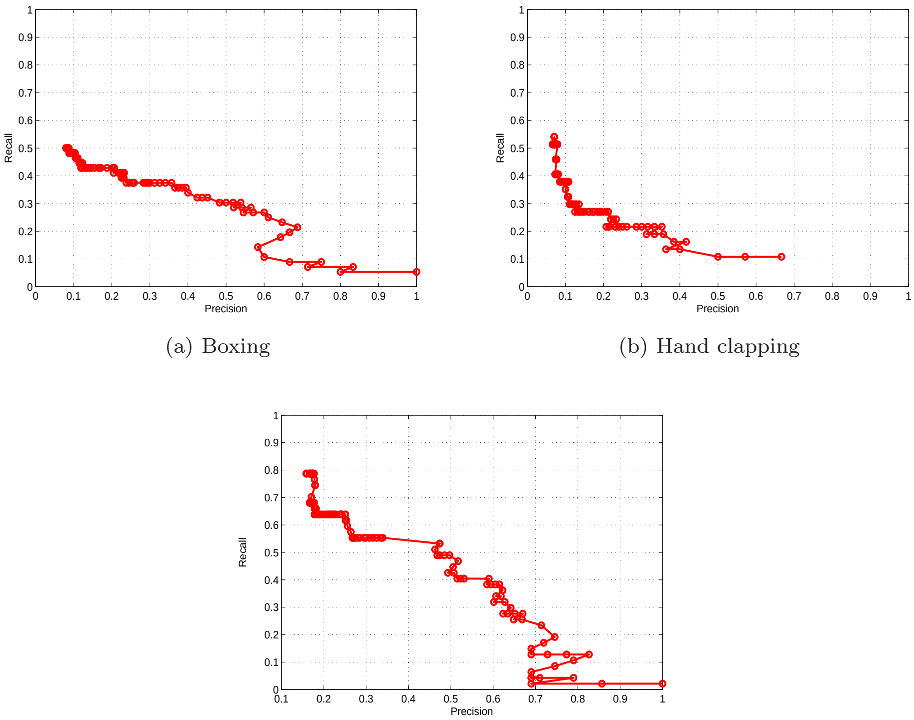
$$Precision = \frac{\sum_{i=1}^{m} H(Q_i^g)}{n} \tag{7.19}$$

$$Recall = \frac{\sum_{j=1}^{n} T(Q_j^d)}{m} \tag{7.20}$$

Varying the threshold on the detection score ROC curve can be obtain for three actions in MSR dataset (See Figure 7.4). We compare average precision (AP) of the three actions in MSR with other state-of-the-art approaches in Table 7.1. We obtain higher AP in *boxing* and *hand waving* actions.

7.3.2 Large scale activity detection

For this experiment 66 videos are used where the ground truth annotation and *scoring software* are available. To prune the search space we apply the region extraction algorithm (Section 7.1). Region pruning algorithm (Algorithm 9) extracts $\sim 3K$ candidate regions from each video. To validate the candidate region we perform a recall test in a similar way described in Equation 7.18 with δ_2 set to 0.75. So, 75% overlap with the ground truth is considered as *hit*. Table 7.2 shows the number of *hit* w.r.t the ground truth. We obtain quite high recall in each category. Our



(a) Boxing

(b) Hand clapping

(c) Hand waving

Figure 7.4: ROC curves of the three actions, (a) boxing, (b) hand clapping and (c) hand waving of MSR dataset.

method outperforms the tracking based region extraction algorithm [138], where first tracking is used and then tracks are divided into units of 3-4 seconds segments (with 2 seconds overlap) each, resulting more than $20K$ detection units. This approach fails to detect the activity that are happening during longer duration than the detection units. On the other hand, as mentioned before, our region pruning method extracts on an average $3K$ regions per video yet obtained a high recall rate.

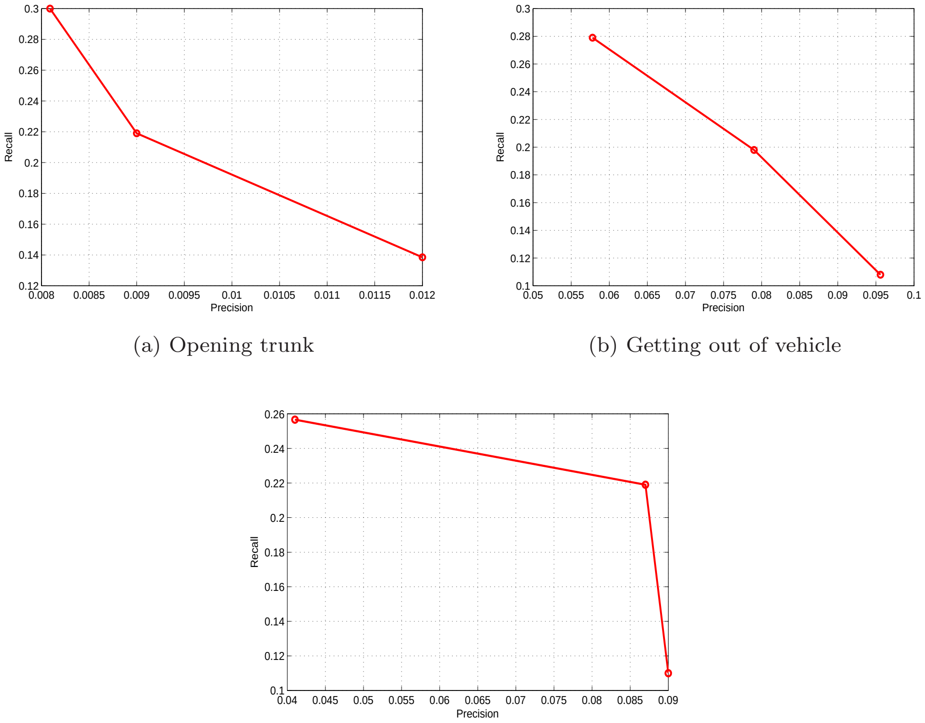
To evaluate the performance of our generalized hough transform approach we use the *scoring software* and generate ROC curves for different activities. Due to high computational time, we could compute ROC curves of three activities, *getting out of vehicle*, *getting into vehicle* and *opening trunk*. These results are obtained using a N-fold cross validation setup. In particular, for each of these activities 5 videos are randomly chosen for test. Figure 7.5 shows the obtained ROC curves. Event recognition in VIRAT dataset is extremely difficult which is reflected from the obtained low precision rate. We perform best in the “getting out of the vehicle” and similar performance in “getting into the vehicle” action can be observed.

Table 7.2: Comparison of recall test performed in the VIRAT dataset. Proposed region extraction algorithm outperforms the tacking based method of [138] to identify initial motion regions where activity of interest may present. Events categories are, (1) loading, (2) unloading, (3) opening trunk (4) closing trunk, 5 getting into vehicle and (6) getting out of vehicle.

Event category	1	2	3	4	5	6
# Ground truth	11	16	18	19	61	63
Our approach	10	15	16	19	60	62
Oh et al. [138]	6	8	8	9	18	14

7.4 Conclusion

In this chapter, we present a novel approach for event detection in large scale activity dataset using max-margin Hough transformation framework. We tackle the large search space by applying a region extraction algorithm which is based on motion segmentation and region clustering. This algorithm is simple and obtains better recall compared to tracking based approaches. For activity detection, generalized Hough transformation technique is applied which is popular in the field of object recognition. We apply a max-margin framework for learning the weights of the visual vocabularies. Finally, a verification SVM is used to obtain the overall score of the detected event hypothesis obtained from Hough transformation framework. To evaluate our approach large scale activity detection dataset is used. We obtain so far the best result on this dataset. To show the effectiveness of our method similar test on small scale benchmark dataset is also performed with state-of-the-art result. More number of tests on the large scale videos would be one of the next steps. Usage of more motion features and improving the vitrification SVM by adding pyramid level and boosting would also be an interesting area to explore.



(a) Opening trunk

(b) Getting out of vehicle

(c) Getting into vehicle

Figure 7.5: ROC curves of the three activities, (a) Opening trunk, (b) getting out of the vehicle and (c) getting into vehicle. We obtain best result in "getting out of vehicle" activity.

Chapter 8

Conclusions and future work

"In my end is my beginning."

by T.S. Eliot

In this chapter a summary of the thesis is presented by revisiting the main modules and contributions. We analyze the strengths and weaknesses of the proposed methods. Finally, a brief overview of the future research possibilities in the area of action, activity recognition is also discussed.

Throughout this thesis several methods of 'action' and 'activity' recognition is presented. This chapter summarizes each main chapters by revisiting the contributions, strengths and weaknesses of the proposed methods.

8.1 Summary and contributions of the proposed action recognition techniques

In thesis we present a series of methods for visual event recognition from video. Chapter 1 introduces the main concept of action recognition in video. Here we also specify the scope of our research work as the field of visual event recognition in video is a vast research area in itself.

In the Chapter 2, a brief overview of the state-of-the-art methods are presented. Here we define the action recognition taxonomy in relation to our thesis scope, by dividing the state-of-the-art methods into *three* different categories, action recognition using single camera, multi-view action recognition and activity detection. Literature review in each of these categories are presented along with the contributions to the state-of-the-art.

We introduce the first approach to action recognition by using an ensemble of body-part based probabilistic model in Chapter 3. The main contribution in this work is to introduce a novel strong model based approach can able to distinguish between similar actions by only considering the body parts having major contribution to the action. To perform this task HMM is used to model the stochastic movement of the body-parts. We present a view-point invariant human detection and example-based body part detectors to model the characteristic human poses as they execute various actions.

Chapter 4 presents an action recognition framework to utilize the concept of key-poses extraction for human action. We apply a real time human detector using ASEF and add a verification SVM to avoid the with the scale problem of the ASEF. The main contribution of this work is to show that the key-poses extracted from the limb dynamics features can be used to represent an action as a Bag-of-key poses model. We introduce the use of directional HOG to model the limb dynamics, i.e. to model the important pose variations during an action by using direction HOG features.

The limitations of the above to methods are mainly the application of model based approach. Action recognition community is moving towards more and more complex actions which are often recorded in real scenario having very low resolution and totally unconstrained environment. In these cases, human detectors and part detector may not work at their best and eventually rest of the system would suffer, resulting in low performance.

To overcome this issue, next chapter (Chapter 5), discusses an in depth analysis in the field of spatio-temporal interest point for action recognition. Here the key contribution is to introduce the inhibition mechanism of the oriented neurons of visual cortex to suppress unwanted back-ground STIPs detected due to background clutter and camera motion. We also propose a novel Bag-of-video words model, where we combine spatial pyramid and vocabulary compression techniques, resulting in a compact action representation for robust action recognition. This method is successfully applied to the datasets having actions in complex scenarios.

Chapter 6 extends the concept of selective STIPs describe in the previous chapter in multi-camera framework. Here we propose a novel 4D STIPs for action recognition in multi-camera systems. The main motivation is to address the use of model free approaches in 3D action recognition like in 2D counterpart. We also introduce a novel view-invariant histogram of 3D optical flow feature for action representation.

Finally, we dedicate the research work to continuous visual event recognition in large scale dataset in the Chapter 7. Here we have adapted the max-margin Hough Transformation technique used in 2D object detection in the activity detection. This work is an attempt to enter into a very challenging domain of action recognition and it as well opens up different future research directions.

8.2 Future work

Through this thesis work various approaches to action recognition, from strong model based to model free, are proposed. Different contributions and limitations of each of these methods are actually widening the future research directions.

8.2.1 Joining the model based and model-free approach:

The first line of future work would introduce the joining of model based and model free approach. In particular, a part based human detection along with the STIPs can handle complex actions more robustly. In this case, we may take the advantages of part-based and local feature based approaches for more improved action recognition. In this case, part based human detector will contribute stochastic movement of body parts, while STIPs will contribute the local motion model. A combination of HMM and Bag-of-visual words with SVM could finally results in an efficient human action recognition system.

8.2.2 Understanding different motion regions in video:

Understanding and modeling motion in video is still an open problem of Computer Vision. We have faced the problem of camera motion in video while performing STIPs in complex, unconstrained videos. Towards this end, introduction of an algorithm to identify different motion part of a video would be a interesting area to revisit. In particular, a Gabor filter could be used as an useful tool in this regard. Depending on the frequency component of the 1D Gabor filter, the motion regions are located in temporal direction. So, to localize all possible motion frequency, we should vary the frequency component of the Gabor filter up to infinity. This has a practical limitation. Our idea is to introduce kernel theory to handle the infinite Gabor frequency representation which will result in a kernel Gabor filter bank for localizing different frequency component. Now, to group the similar motion components, we would like to explore the mutual information via entropy.

8.2.3 Use of depth information for action recognition:

To extended the work of human action recognition in multi-camera systems, the obvious choice is to use time-of-flight cameras or Kinect cameras to obtain directly the depth information. Depth information will provide a different 4D STIPs but finding the corner points directly using the spatial and depth information. Extension of action recognition methods to gestures recognition using Kinect and using gestures as multi-modal interaction can be very useful for the applications like intelligent gaming and functional rehabilitation.

8.2.4 Exploring new features and representation for action recognition:

During the research work, we have encountered a number of features like, optical flow, SIFT, SURF, N-jet, HOG3D and HOF for action recognition. Although these features are nicely capturing the global and local characteristics of the action, the need of new features can be avoided. To this end, extension of the features applied in 2D object recognition task is an useful direction to explore. In this regards, local binary pastern used in 2D can be extended to 3D to capture the local spatio-temporal texture for action. For object recognition, GIST features [166, 165] are gaining popularity, since it can robustly capture the object and scene description globally. Extension of GIST like feature in action recognition could improve the result.

For action representation, the most popular approach is Bag-of-video words model. But the research towards the introduction of graph based method for action representation is till in its inception. Representing different actions by using graphs could be a very interesting idea, then action detection and recognition task would be considered as a graph matching problem.

8.2.5 Applying outdoor surveillance to indoor scenario:

Our method of outdoor surveillance can be extended for an indoor surveillance, where the system can train to detect certain activities or abnormal behaviour. This system is very important for health care application, for example, the smart elderly care system. Another extension is to use Kinect camera network inside our framework and to work using depth image. This is an important criteria for the indoor surveillance to maintain the privacy of the users. Incorporation of the scene modeling framework in the activity recognition could also be considered as an interesting research direction and it can help monitoring the behaviour of the users and avoid false alarm.

Appendix A

Action recognition datasets

[Action recognition datasets] Following are the benchmark action recognition datasets on which all the proposed methods are applied.

KTH dataset

This is most common benchmark action recognition dataset introduced in [159][1]. It covers 25 subjects and *four* different recording conditions of the videos, outdoors, outdoors with zooming, outdoors with different clothing and indoors. There are *six* actions in this dataset: *walking*, *running*, *jogging*, *boxing*, *clapping* and *waving* (See Figure A.1). There are considerable variation in the performance and duration, and somewhat in the viewpoint. The backgrounds are static. Apart from the zooming scenario, there is only slight camera movement. The videos present in this dataset are of 120 × 160

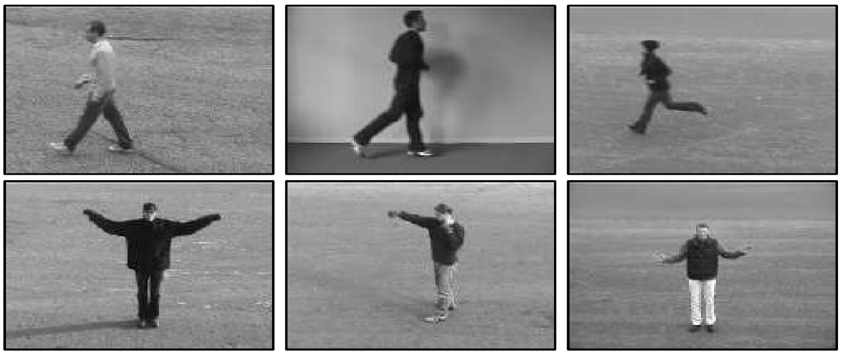


Figure A.1: Sample image frames from KTH action dataset [159].

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

Weizmann dataset

This is a popular benchmark dataset presented in [63][2]. This contains 90 low-resolution (180 × 144) videos separated into *ten* actions performed by *nine* persons. The actions are *bend*, *jumping-jacks*, *jump*, *jumping-in-place*, *run*, *gallop-sideway*, *skip*, *walk*, *one-hand-waving*, *two-hands-waving*. The backgrounds are static and foreground silhouettes are included in the dataset. All the videos are recorded in a static view point. Figure A.2 shows the image frames from different actions present in this dataset.

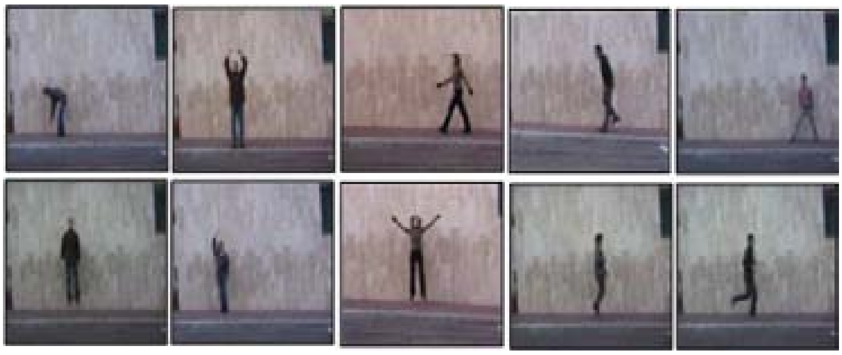


Figure A.2: Sample image frames from Weizmann action dataset [63].

CVC action dataset

This is a semi-complex action recognition dataset introduced in [27, 29][3]. It contains *seven* actions: *walking*, *jogging*, *running* (all these actions are performed in horizontal and vertical tow-way paths), *hand-waving*, *two-hands-waving*, *jump-in-place* and *bending*. These actions are performed by *five* different actors. This dataset is recorded using fixed camera and is has textured background. The video frame resolution 640 × 480 and 320 × 240 are available. Image frame samples from different actions in this dataset are shown in Figure A.3.

Youtube action dataset

The YouTube dataset [112][4] is a collection of 1168 complex and challenging YouTube videos of 11 human actions categories: *basketball shooting*, *volleyball spiking*, *trampoline jumping*, *soccer juggling*, *horseback riding*, *cycling*, *diving*, *swinging*, *golf swinging*, *tennis swinging* and *walking (with a dog)*. The dataset has the following properties: a mix of steady cameras and shaky cameras, cluttered background, low resolution,

[2] http://www.wisdom.weizmann.ac.il/∼vision/SpaceTimeActions.html

[3] http://iselab.cvc.uab.es/files/Tools/CvcActionDataSet/index.htm

[4] http://www.cs.ucf.edu/∼liujg/YouTube_Action_dataset.html

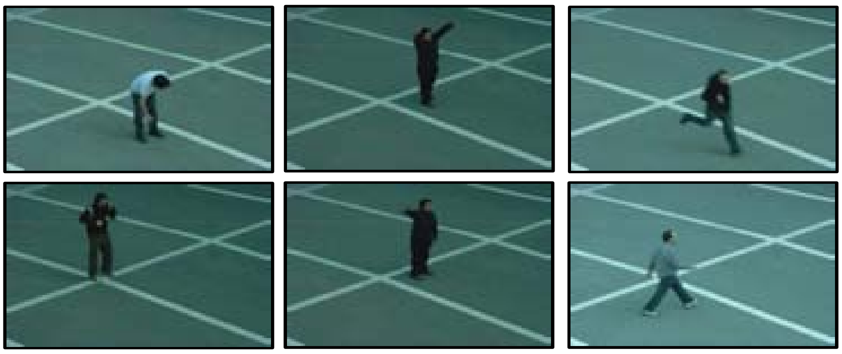


Figure A.3: Sample image frames from CVC action dataset [27, 29]. This dataset has KTH like actions with a textured background.

and variation in object scale, viewpoint and illumination. The first four actions are easily confused with jumping, the next two may have similar camera motion, and all the swing actions share some common motions. Some actions are also performed with objects such as a horse, bike or dog. Figure A.4 gives an overview of the image samples of this dataset.

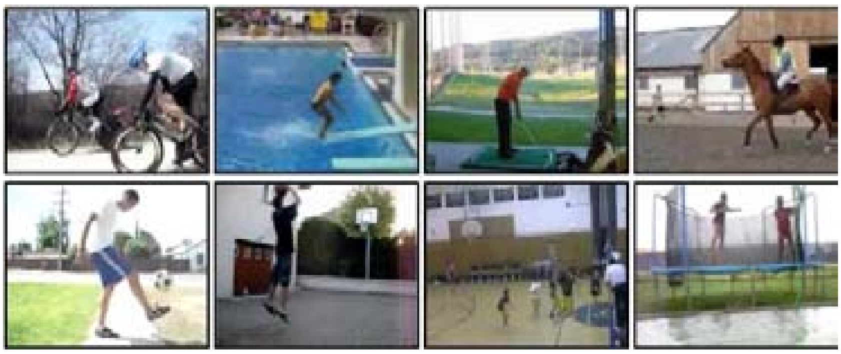


Figure A.4: Sample image frames from Youtube action dataset [112]. This dataset is very challenging with low resolution and camera motion.

Hollywood 2 dataset

The Hollywood 2 dataset [125][5] is composed of video clips extracted from 69 Hollywood movies, and contains 12 classes of human actions and 10 classes of scenes distributed over 3669 video clips, resulting in approximately 20.1 hours of video in total (See Figure fig:h2ActionDataset). The 12 actions are: *AnswerPhone*, *DriveCar*, *Eat*, *FightPerson*, *GetOutCar*, *HandShake*, *HugPerson*, *Kiss*, *Run*, *SitDown*, *SitUp* and *StandUp*. In total, there are 1707 action samples divided into a training set

[5] http://www.di.ens.fr/∼laptev/actions/hollywood2/

(823 sequences) and a test set (884 sequences), where train and test sequences are obtained from different movies. The dataset contains approximately 150 samples per action class and 130 samples per scene class in training and test subsets, and intends to provide a comprehensive benchmark for human action recognition in realistic and challenging settings.

Figure A.5: Image frames collected from Hollywood 2 action dataset [125]. This dataset is obtained from different movies and contains actions in real scenario and camera motion.

HumanEva dataset

This dataset is introduced in [167][6]. It contains 7 calibrated video sequences (4 grayscale and 3 colour) that are synchronized with $3D$ body poses obtained from a motion capture system. The dataset contains 4 subjects performing 6 common actions: *walking*, *jogging*, *boxing*, *gesture*, *combo* and *throw catch*. But, the action *combo* and *walking* are the same so, we combine them as *walking* action and use 5 actions for the action recognition. Figure A.6 shows the image frames taken from this dataset.

i3DPost action dataset

The i3DPost dataset [60][7] consists of 8 actors performing 10 different actions, where 6 are single actions: *walk*, *run*, *jump*, *bend*, *hand-wave* and *jump-in-place*, and 4 are combined actions: *sit-stand-up*, *run-fall*, *walk-sit* and *run-jump-walk*. The subjects have different body sizes, clothing and are of different sex and nationalities. The multi-view videos have been recorded by a 8 calibrated and synchronized camera setup in high definition resolution (1920 × 1080), resulting in a total of 640 videos. For each video frame a 3D mesh model of relatively high detail level ($20,000 - 40,000$ vertices and $40,000 - 80,000$ triangles) of the actor and the associated camera calibration parameters are available. The mesh models were reconstructed using a global

[6] http://vision.cs.brown.edu/humaneva/

[7] http://kahlan.eps.surrey.ac.uk/i3dpost_action/

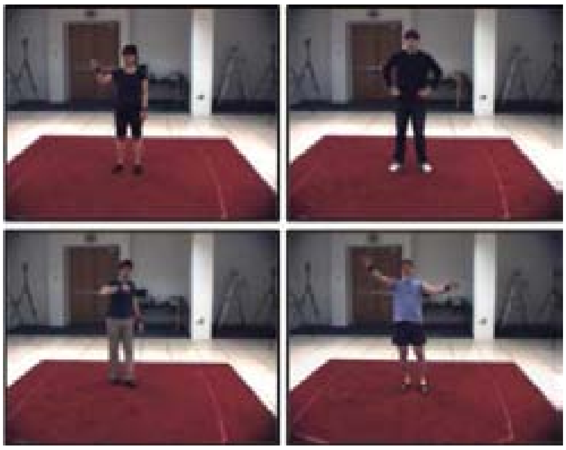


Figure A.6: Sample image frames collected from HumanEva action dataset [167].

optimization method proposed by Starck et al.[173]. Figure A.7 shows multi-view actor/action images and 3D mesh model examples from the i3DPost dataset.

IXMAS action dataset

The IXMAS dataset[8] is introduced in [200]. It consists of 12 non-professional actors performing 13 daily-life actions 3 times. The actions are: *check watch*, *cross arms*, *scratch head*, *sit down*, *get up*, *turn around*, *walk*, *wave*, *punch*, *kick*, *point*, *pick up* and *throw*. The dataset has been recorded by 5 calibrated and synchronized cameras, where the actors chose freely position and orientation, and consists of image sequences (390 × 291) and reconstructed 3D volumes (64 × 64 × 64 voxels), resulting in a total of 2340 action instances for all 5 cameras. Compared to i3Dpost the IXMAS dataset is of lower data quality and resolution. Figure A.8 shows sample image frames and 3D volume of this dataset.

CMU acton dataset

The CMU dataset is composed of 48 video sequences of five action classes: *jumping-jacks*, *pick-up*, *push-button*, *one-hand-waving* and *two-hands-waving*. The test data contains 110 videos (events) which are down-scaled to 160 × 120 in resolution. This dataset has been recorded by a hand-held camera with moving people and vehicles in the background, and is known to be very challenging.

[8] http://4drepository.inrialpes.fr/public/viewgroup/6

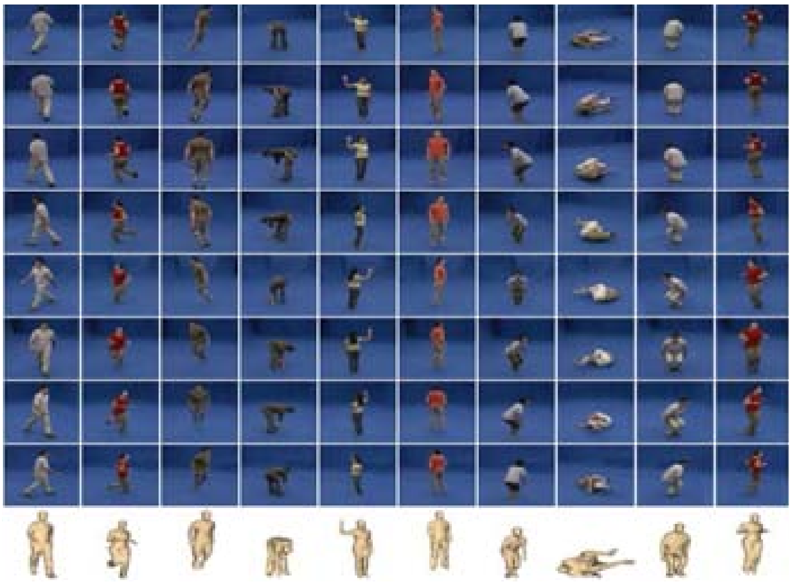


Figure A.7: Image and 3D mesh model examples for the 10 actions from the i3DPost Multi-View Human Action Dataset. The columns correspond to the 10 different actions performed by the 8 actors, where the first 6 columns show the single actions and the last 4 columns show the combined actions. The first 8 rows depict images captured from the 8 camera views, while the 9^{th} row shows the corresponding 3D mesh models.

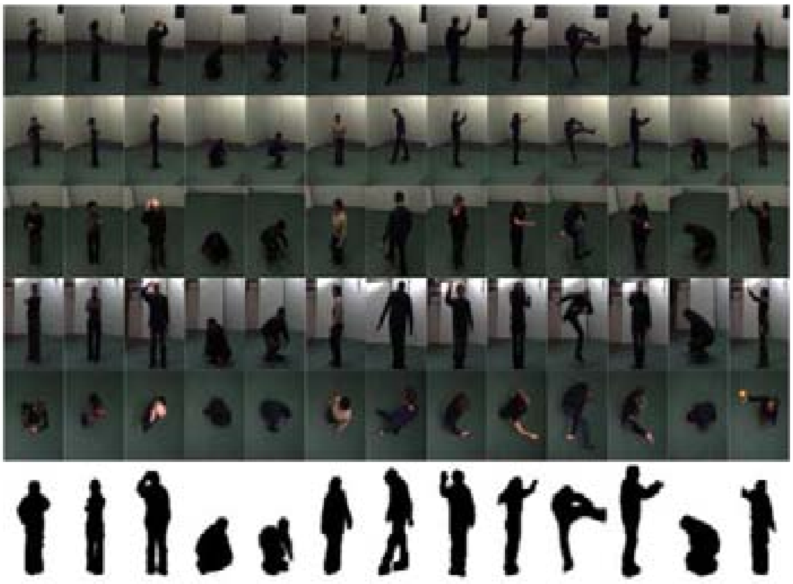


Figure A.8: Image and 3D voxel-based volume examples for the 13 actions from the IXMAS Multi-View Human Action Dataset[200].The columns correspond to the 13 different actions performed by the 12 actors. The first 5 rows depict images captured from the 5 camera views, while the 6^{th} row shows the corresponding 3D volumes.

MSR action dataset

The Microsoft research action dataset (MSR)[9] consists of 54 video sequences recorded in a crowded environment [217] (See Figure A.9). It has total of 203 actions: *hand-clapping*, *hand-waving* and *boxing*, performed by 10 subjects. The sequences contain multiple types of action recorded in indoor and outdoor scenes with cluttered and moving backgrounds. Some sequences contain multiple actions performed by different people. Each video is of low resolution (320 × 240) with a frame rate of 15 frames per second, and their lengths are between 32 to 76 seconds.

HERMES indoor dataset

This dataset[10] is described in [62]. In this HERMES sequence (2003 frames @ 15 fps, 1392 × 1040 pixels) there are three people in a room. They act in a discussion sequence sitting around a table, where bags are carried, left and picked from the floor, and bottles are carried, left and picked from the vending machine and from the table. In this discussion sequence several agents are involved in different simultaneous grouping, grouped and splitting events, while they are partially or completely occluded. Figure A.10 shows some sample image frames for this dataset.

[9] http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/default.htm

[10] http://iselab.cvc.uab.es/indoor-database

Figure A.9: Different actions of MSR action dataset [217]. This dataset contains actions in crowded scenes both in indoor and outdoor scenario

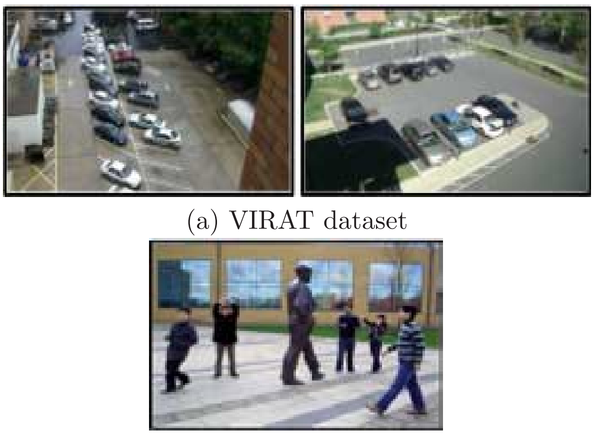
Figure A.10: Image frames from HERMES indoor dataset [62].

VIRAT activity dataset

In our experiments we use the Release 1.0 [11] [138] of the dataset which was publish in the CVPR'11 activity recognition challenge [12]. It contains 66 videos as *training set* with available ground truth annotation and *scoring software*. The *training set* contains 3 scenes (Figure A.11.a). Besides, in the above mentioned activity recognition challenge, 128 videos were latter released as *test videos* where total 6 scenes are present, three same scenes like in *training set* with three additional scenes (see Figure). The *test set* does not have *scoring software* or ground truth annotations. These videos are captured by stationary HD cameras (1080p or 720p). Some of them contains slight jitter due to environmental condition. Heights of humans within videos range 25 $\sim$ 200 pixels, constituting $2.3 - 20\%$ of the heights of recorded videos with

[11] $http : //www.viratdata.org/virat/virat_a rchive1.html$

[12] $http : //www.umiacs.umd.edu/conferences/cvpr2011/ARC/$



(a) VIRAT dataset

(b) Multi-KTH dataset

Figure A.11: Image frame examples from (a) VIRAT activity recognition dataset and (b) Multi-KTH dataset[183].

average being about 7%. There are total 6 activities, i) *person loading an object to a vehicle*, ii) *person unloading an object from a vehicle*, iii) *person opening a vehicle trunk*, iv) *person closing a vehicle trunk*, v) *person getting inside into a vehicle* and vi) *person getting out of a vehicle*. This dataset is extremely challenging.

Multi-KTH action dataset

The Multi-KTH dataset is a more challenging version of the KTH dataset and is introduced in [183]. It contains 5 (except *running*) of the 6 KTH-actions, which have been recorded by a hand-held camera, with multiple simultaneous actors, a significant amount of camera motion, scale changes and a more realistic cluttered background. Figure A.11.b shows the sample image frames taken from this dataset.

Appendix B

Publications

Refereed journals

- Bhaskar Chakraborty, Michael B. Holte, Thomas B. Moeslund and Jordi Gonzàlez. Selective Spatio-Temporal Interest Points. *Computer Vision and Image Understanding, Elsevier*, 116(3), pages: 396-410. 2012.

- Michael B. Holte, Bhaskar Chakraborty, Jordi Gonzàlez and Thomas B. Moeslund, A Local 3D Motion Descriptor for Multi-View Human Action Recognition from 4D Spatio-Temporal Interest Points. *IEEE Journal of Selected Topics in Signal Processing*. In press. 2012..

- Bhaskar Chakraborty, Andrew D. Bagdanov, Jordi Gonzàlez and F. Xavier Roca, Human action recognition using an ensemble of body-part detectors. *Expert Systems, The Journal of Knowledge Engineering, Wiley-Blackwell*. In press. September, 2011.

Refereed major conferences

- Bhaskar Chakraborty, Michael B. Holte, Thomas B. Moeslund, Jordi Gonzàlez and F. Xavier Roca, A Selective Spatio-Temporal Interest Point Detector for Human Action Recognition in Complex Scenes. *ICCV'11:* 13th *International Conference on Computer Vision*, Barcelona, Spain. November, 2011.

- Bhaskar Chakraborty, Andrew D. Bagdanov and Jordi Gonzàlez, Towards Real-Time Human Action Recognition. *IbPRIA'09:* 4th *Iberian Conference on Pattern Recognition and Image Analysis*, Póvoa do Vergim, Portugal, pages 425-432. June, 2009.

- Bhaskar Chakraborty, Ognjen Rudovic and Jordi Gonzàlez, View-Invariant Human-Body Detection with Extension to Human Action Recognition using Component-Wise HMM of Body Parts. *FG'08:* 8th *IEEE International Conference on Au-*

tomatic Face and Gesture Recognition, Amsterdam, The Netherlands, pages 1-6. September, 2008.

- Bhaskar Chakraborty, Marco Pedersoli, Jordi Gonzàlez, View-invariant human action detection using component-wise HMM of body parts. *AMDO'08: 5th International Workshop on Articulated Motion and Deformable Objects*, Andratx, Mallorca, Spain, pages: 208-217. July, 2008.

- Marco Pedersoli, Jordi Gonzàlez, Bhaskar Chakraborty and Juan Jose Villanueva, Enhancing Real-time Human Detection based on Histograms of Oriented Gradients, *CORES'07:* 5th *International Conference on Computer Recognition Systems*, Wroclaw, Poland, pages 739-746. October, 2007.

Technical reports

- Bhaskar Chakraborty. Human action recognition in complex scenes using a selective STIP detector. *CVCRD'11: State of the art of Research and Development.*

- Bhaskar Chakraborty, Andrew D. Bagdanov, Jordi Gonzàlez. Interest point based human action recognition. *CVCRD'09: Progress of Research & Development.*

- Bhaskar Chakraborty, Ognjen Rudovic, Mikhail Mozerov, Jordi Gonzàlez. Towards Real time Human Action Recognition Using Correlation Approach. *CVCRD'08: Current Challenges in Computer Vision.*

- Bhaskar Chakraborty, Ignasi Rius, Marco Pedersoli, Mikhail Mozerov, Jordi Gonzàlez. Component-based human detection. *CVCRD'07: Computer Vision: Advances in Research & Development.*

Projects

- **VIDI-Video IST 045547**: Interactive Semantic Video Search using Machine-Learned Audio-Visual Concepts.

- **HERMES IST 027110**: Human-Expressive Representations of Motion and their Evaluation in Sequences.

- **CONSOLIDER-INGENIO 2010 MIPRCV**: Multimodal interaction in pattern recognition and computer vision.

- **CICYT ERINYES TIN2009-14501-C02**: Epistemological reasoning to interpret context and security events for surveillance.

- **CICYT SISYPHUS TIN2006-14606**: Security indoor system for places with human in scenes.

Bibliography

[1] J. K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images - a review. *Proceedings of the IEEE*, 76(8):917–935, 1988.

[2] J.K. Aggarwal and M.S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 43(3):1–43, 2011.

[3] Md.A.R. Ahad, T. Ogata, J.K. Tan, H. Kim, and S. Ishikawa. Motion recognition approach to solve overwriting in complex actions. In *FG'08: Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pages 1–6, 2008.

[4] M. Ahmad and S.W Lee. Hmm-based human action recognition using multiview image sequences. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, volume 1, pages 263–266, 2006.

[5] A. Ali and J.K. Aggarwal. Segmentation and recognition of continuous human activity. In *DREV'01: IEEE Workshop on Detection and Recognition of Events in Video*, pages 28–35, 2001.

[6] S. Ali, A. Basharat, and M. Shah. Chaotic invariants for human action recognition. In *ICCV'07: Proceedings of the IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[7] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 32(2):288–303, 2010.

[8] M. Ankerst, G. Kastenmüller, H.P. Kriegel, and T. Seidl. 3d shape histograms for similarity search and classification in spatial databases. In *ISASD'99: Proceedings of the 6th International Symposium on Advances in Spatial Databases*, pages 207–226, 1999.

[9] D. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Patteren Recognition*, 13(2):111–122, 1981.

[10] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV'08: Proceedings of the Fourth European Conference on Computer Vision*, 2006.

[11] S. Baysal, M.C. Kurt, and P. Duygulu. Recognizing human actions using key poses. In *ICPR'10: Proceedings of the International Conference on Pattern Recognition*, pages 1727–1730, 2010.

[12] P. Beaudet. Rotationally invariant image operators. In *ICPR'78: Proceedings of the 4th International Joint Conference on Pattern Recognition*, pages 579–583, 1978.

[13] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[14] M. Black and Y. Yacoob. Parameterized modeling and recognition of activities. *Computer Vision and Image Understanding*, 73:232–247, 1999.

[15] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space time shapes. In *ICCV'05: Proceedings of the IEEE International Conference on Computer Vision*, pages 1395–1402, 2005.

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

[17] D.S. Bolme, B.A. Draper, and J.R. Beveridge. Average of synthetic exact filters. In *CVPR'09: Proceedings of IEEE Computer Society Conference on Computer Vision*, pages 2105–2112, 2009.

[18] D.S. Bolme, Y.M. Lui, B.A. Draper, and J.R. Beveridge. Simple real-time human detection using a single correlation filter. In *IWPETS'09: Proceedings of International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 2105–2112, 2009.

[19] M. Bregonzio, Shaogang Gong, and Tao Xiang. Recognizing action as clouds of space-time interest points. In *CVPR'09: Proceedings of the IEEE Computer Society Conference on*, 2009.

[20] M. Bregonzio, J. Li, S. Gong, and T. Xiang. Discriminative topics modelling for action feature selection and recognition. In *BMVC'10: Proceedings of the British Machine Vision Conference*, 2010.

[21] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[22] L. Bretzner and T. Lindeberg. Feature tracking with automatic selection of spatial scales. *Computer Vision and Image Understanding*, 71(3):385–392, 1998.

[23] N. Buch, J. Orwell, and S.A. Velastin. 3d extended histogram of oriented gradients (3dhog) for classification of road users in urban scenes. In *BMVC'09: Proceedings of the British Machine Vision Conference*, 2009.

[24] C. Canton-Ferrer, J.R. Casas, and M. Pardás. Human model and motion based 3d action recognition in multiple view scenarios. In *EUSIP'06: Proceeding of the 14th European Signal Processing Conference*, 2006.

[25] L. Cao, Z. Liu, and T.S. Huang. Cross-dataset action detection. In *CVPR'10: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[26] S. Carlsson and J. Sullivan. Action recognition by shape matching into key frames. In *WMECV'01: Proceeding of the Workshop on Models versus Exemplars in Computer Vision*, 2006.

[27] B. Chakraborty, M.B. Holte and. T.B. Moeslund, and J. Gonzáles. A selective spatio-temporal interest point detector for human action recognition in complex scenes. In *ICCV'11: Proceedings of the IEEE International Conference on Computer Vision*, 2011.

[28] B. Chakraborty, A.D. Bagdanov, and J. Gonzlez. Towards real time human action recognition. In *IbPRIA'09: Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis*, pages 425–432, Andratx, Mallorca, Spain, 2009.

[29] B. Chakraborty, M.B. Holte, T.B. Moeslund, and J. Gonzàlez. Selective spatio-temporal interest points. *Computer Vision and Image Understanding*, 116(3):396–410, 2012.

[30] B. Chakraborty, O. Rudovic, and J. Gonzlez. View-invariant human action detection using component-wise hmm of body parts. In *AMDO'08: Proceedings of the V Conference on Articulated Motion and Deformable Objects*, pages 208–217, Andratx, Mallorca, Spain, July 2008.

[31] C-C. Chang and C-J. Lin. *LIBSVM: a library for support vector machines*, 2001.

[32] R. Chaudhry, A. Ravichandran, G.D. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *CVPR'09: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1932–1939, 2009.

[33] H-S. Chen, H-T. Chen, Y-W. Chen, and S-Y. Lee. Human action recognition using star skeleton. In *VSSN'06: Proceedings of the ACM international workshop on Video surveillance and sensor networks*, pages 171–178, New York, NY, USA, 2006. ACM.

[34] S. Cherla, K. Kulkarni, A. Kale, and V. Ramasubramanian. Towards fast, view-invariant human action recognition. In *CVPR Workshops'08: Proceedings of the Workshop of IEEE Computer Society Conference on Computer Vision*, 2008.

[35] I. Cohen and H. Li. Inference of human postures by classification of 3d human body shape. In *AMFG'03: Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, 2003.

[36] D. Comaniciu and P. Meer. Mean shift: A robust approach towards feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 24:603–619, 2002.

[37] C. Cortes and V. Vapnik. Support -vector networks. *Machine Learning*, 20(3):273–297, 1995.

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

[39] R. Cutler and M.Turk. View-based interpretation of real-time optical flow for gesture recognition. In *ICAFGR'98: Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pages 416–421, 1998.

[40] N. Dalal and B.Triggs. Histograms of oriented gradients for human detection. In *CVPR'05: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 01, pages 886–893, Washington, DC, USA, June 2005. IEEE Computer Society.

[41] S. Danafar and N. Gheissari. Action recognition for surveillance applications using optic flow and svm. In *ACCV'07: Proceedings of the Asian Conference on Computer Vision*, pages 457–466, 2007.

[42] J.W. Davis and S.R. Taylor. Analysis and recognition of walking movements. In *ICPR'04: Proceedings of the International Conference on Pattern Recognition*, pages 315–318, 2002.

[43] M. Dikmen. Surveillance event detection. In *In TRECVID Video Evaluation Workshop*, 2008.

[44] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VSPETS'05: Proceedings of the 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005.

[45] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[46] A.A. Efros, A.C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV'03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, Washington, DC, USA, 2003. IEEE Computer Society.

[47] A.M. Elgammal, V.D. Shet, Y. Yaccob, and L.S. Davis. Learning dynamics for exemplar-based gesture recognition. In *CVPR'09: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 571–578, 2003.

[48] J. Fan, Y. Wu, and S. Dai. Discriminative spatial attention for robust tracking. In *ECCV'10: Proceedings of the European Conference on Computer Vision*, pages 480–493, 2010.

[49] A. Farhadi and M.K. Tabrizi. Learning to recognize activities from the wrong view point. In *ECCV'08: Proceedings of the European Conference on Computer Vision*, 2008.

[50] A. Fathi and G. Mori. Action recognition by learning mid level motion feature. In *CVPR'08: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[51] P. Fihl and T.B. Moeslund. Invariant gait continuum based on the duty-factor. *Signal Image and Video Processing*, 3(4):391–402, 2008.

[52] J. Gall and V. Lempitsky. Class-specific hough forests for object detection. In *CVPR'09: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[53] J. Gall, A. Yao, N. Razavi, L.J.V Gool, and V.S. Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 33(11):2188–2202, 2011.

[54] D. Gao, S. Han, and N. Vasconcelos. Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:989–1005, 2009.

[55] D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[56] D.M. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *ICCV'99: Proceedings of the IEEE International Conference on Computer Vision*, pages 87–93, 1999.

[57] Z. Ghahramani. An introduction to hidden markov models and bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):9–42, 2001.

[58] A. Gilbert, J. Illingworth, and R. Bowden. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *ICCV'09: Proceedings of the IEEE International Conference on Computer Vision*, 2009.

[59] A. Gilbert, J. Illingworth, and R. Bowden. Action recognition using mined hierarchical compound features. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 33(5):883–897, 2011.

[60] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, and I. Pitas. The i3dpost multi-view and 3d human action/interaction database. In *CVMP'09: Proceedings of the Conference for Visual Media Production*, pages 159–168, 2009.

[61] N. Gkalelis, N. Nikolaidis, and I. Pitas. View independent human movement recognition from multi-view video exploiting a circular invariant posture representation. In *ICME'09: Proceedings of the IEEE international conference on Multimedia and Expo*, pages 394–397, 2009.

[62] J. Gonzàlez, D. Rowe, J. Varona, and F.X. Roca. Understanding dynamic scenes based on human sequence evaluation. *Image and Vision Computing*, 27(10):1433–1444, 2009.

[63] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 29:2247–2253, 2007.

[64] M. Grant and S. Boyd, 2008. cvx: Matlab software for disciplined convex programming.

[65] C. Grigorescu, N. Petkov, and M.A. Westenberg. Contour and boundary detection improved by surround suppression of texture edges. *Image Vision Computing*, 22(8):609–622, 2004.

[66] Y. Guo, G. Xu, and S. Tsuji. Understanding human motion patterns. In *CVIP'94: Proceedings of the 12th IAPR Conference on Computer Vision and Image Processing*, pages 325–329, Jerusalem, Israel, 1994.

[67] D. Han, L. Bo, and C. Sminchisescu. Selection and context for action recognition. In *ICCV'09: Proceedings of the IEEE International Conference on Computer Vision*, 2009.

[68] A. Haq, I. Gondal, and M. Murshed. On dynamic scene geometry for view-invariant action matching. In *CVPR'11: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

[69] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: A real time system for detecting and tracking people. *CVPR'98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 962–969, 1998.

[70] C. Harris and M. Stephens. A combined corner and edge detector. In *AVC'88: Proceedings of the Alvey Vision Conference*, 1988.

[71] M.B. Holte, B. Chakraborty, J. Gonzàlez, and T.B. Moeslund. A local 3d motion descriptor for multi-view human action recognition from 4d spatio-temporal interest points. *Accepted in IEEE Journal of Selected Topics in Signal Processing*, 2012.

[72] M.B. Holte, T.B. Moeslund, N. Nikolaidis, and I. Pitas. 3d human action recognition for multi-view camera systems. In *3DIMPVT'11: Proceedings of the International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 342–349, 2011.

[73] Y. Hu, L. Cao, F. Lv, S. Yan, and T.S. Huang. Action detection in complex scenes with spatial and temporal ambiguities. In *ICCV'09: Proceedings of the IEEE International Conference on Computer Vision*, 2009.

[74] P. Huang and A. Hilton. Shape-colour histograms for matching 3d video sequences. In *ICCV Workshop'09: Proceedings of IEEE 12th International Conference on Computer Vision Workshops*, pages 1510–1517, 2009.

[75] P. Huang, A. Hilton, and J. Starck. Shape similarity for 3d video sequences of people. *Internation Jounrnal of Computer Vision*, 89:362–381, 2010.

[76] P. Huang, J. Starck, and A. Hilton. A study of shape similarity for temporal surface sequences of people. In *3DIM'07: Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling*, 2007.

[77] N. Ikizler, R.G. Cinbis, and P. Duygulu. Human action recognition with line and flow histograms. In *ICPR'08: Proceedings of the 19th International Conference on Pattern Recognition*, pages 1–4, 2008.

[78] N. Ikizler and P. Duygulu. Histogram of oriented rectangles: A new pose descriptor for human action recognition. *Image and Vision Computing*, 27(10):1515–1526, 2009.

[79] A. Iosifidis, N. Nikolaidis, and I. Pitas. Movement recognition exploiting multiview information. In *MMSP'10: Proceedings of the IEEE International Workshop on Multimedia Signal Processing*, pages 427–431, 2010.

[80] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV'07: Proceedings of the Eleventh IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[81] X. Ji and H. Liu. Advances in view-invariant human motion analysis: A review. *IEEE Transactions on Systems, Man, and Cybernetics*, 40(1):13–24, 2010.

[82] Z. Jiang, Z. Lin, and L.S. Davis. A tree-based approach to integrated action localization, recognition and segmentation. In *ECCV Workshops'10: Proceedings of the European Conference on Computer Vision Workshop*, 2010.

[83] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14(2):201–211, 1973.

[84] G. Johansson. Visual motion perception. *Scientific America*, 232(6):76–88, 1975.

[85] A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[86] I.N. Junejo, E. Dexter, I. Laptev, and P. Pérez. Cross-view action recognition from temporal self-similarities. In *ECCV'08: Proceedings of the European Conference on Computer Vision*, 2008.

[87] I.N. Junejo, E. Dexter, I. Laptev, and P. Pérez. View-independent action recognition from temporal self-similarities. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 33(1):172–185, 2011.

[88] I.K. Jung and S. Lacroix. A robust interest points matching algorithm. In *ICCV*, 2001.

[89] M.B. Kaâniche and E.F. Brémond. Gesture recognition by learning local motion signatures. In *CVPR*, 2010.

[90] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *SGP'03: Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 156–164, 2003.

[91] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *ICCV'05: Proceedings of the IEEE International Conference on Computer Vision*, volume 1, pages 166–173, October 2005.

[92] V. Kellokumpu, G. Zhao, and M. Pietikäinen. Human activity recognition using a dynamic texture based method. In *BMVC'08: Proceedings of the British Machine Vision Conference*, 2008.

[93] V. Kellokumpu, G. Zhao, and M. Pietikäinen. Recognition of human actions using texture descriptors. *Machine Vision Application*, 22(5):767–780, 2011.

[94] T.K. Kim, S.F. Wong, and R. Cipolla. Tensor canonical correlation analysis for action classification. In *CVPR'07: Proceedings of the IEEE Computer Society Conference of Computer Vision and Pattern Recognition*, 2007.

[95] A. Kläser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC'08: Proceedings of the British Machine Vision Conference*, 2008.

[96] J.J. Koenderink and A.J. Van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[97] M. Körtgen, M. Novotni, and R. Klein. 3d shape matching with 3d shape contexts. In *CESCG'03: Proceedings of the 7th Central European Seminar on Computer Graphics*, pages 12–19, 2003.

[98] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *CVPR'10: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[99] C. Lampert, M. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient sub-window search. In *CVPR'08: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[100] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2/3):107–123, 2005.

[101] I. Laptev, B. Caputo, C. Schüldt, and T. Lindeberg. Local velocity-adapted motion events for spatio-temporal recognition. *International Journal of Computer Vision*, 108(3):207–229, 2007.

[102] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV'03: Proceedings of the International Conference on Computer Vision*, 2003.

[103] I. Laptev and T. Lindeberg. Local descriptors for spatio-temporal recognition. In *SCVMA'04: Proceedings of the First International Workshop on Spatial Coherence for Visual Motion Analysis*, 2004.

[104] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR'08: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[105] I. Laptev and P. Perez. Retrieving actions from movies. In *ICCV'07: Proceedings of the International Conference on Computer Vision*, 2007.

[106] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3):259–289, 2008.

[107] J. Leiblt, C. Schimd, and K. Schertler. View-point independent object class detection using 3d feature maps. In *CVPR'08: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[108] Z. Lin, L.S. Davis, D. Doermann, and D. Dementhon. Hierarchical part-template matching for human detection and segmentation. In *ICCV'07: Proceedings of the Eleventh IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, October 2007.

[109] Z. Lin, Z. Jiang, and L.S. Davis. Recognizing actions by shape-motion prototype trees. In *ICCV'09: Proceedings of the IEEE 12th International Conference on Computer Vision*, 2009.

[110] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.

[111] J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. In *CVPR'08: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[112] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *CVPR'09: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.

[113] J. Liu and M. Shah. Learning human actions via information maximization. In *CVPR'08: Proceedings of the IEEE Computer Society Conference Computer Vision and Pattern Recognition*, 2008.

[114] J. Liu, M. Shah, B. Kuipers, and S. Savarese. Cross-view action recognition via view knowledge transfer. In *CVPR'11: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

[115] J. Liu, Y. Yang, and M. Shah. Learning semantic visual vocabularies using diffusion distance. In *CVPR'09: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.

[116] C.H. Lo and H.S. Don. 3-d moment forms: Their construction and application to object identification and positioning. *IEEE Transaction of Pattern Analysis and Machine Intelligence*, 11(10):1053–1064, 1989.

[117] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[118] W-L Lu and J.J. Little. Simultaneous tracking and action recognition using the pca-hog descriptor. In *CRV'06: Proceedings of the Canadian Conference on Computer and Robot Vision*, pages 6–13, 2006.

[119] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Imaging Understanding Workshop*, 1981.

[120] Y.M. Lui, J.R. Beveridge, and M. Kirby. Action classification on product manifolds. In *CVPR'10: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[121] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *CVPR'07: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[122] V. Mahadevan and N. Vasconcelos. Spatiotemporal saliency in dynamic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:171–177, 2010.

[123] S. Maji and J. Malik. Object detection using a max-margin hough transform. In *CVPR'09: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[124] K.V. Mardia and R.E Juppe. *Directional Statistics.* Wiley, 2 edition, January 1999.

[125] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR'09: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.

[126] P. Matikainen, M. Hebert, and R. Sukthankar. Representing pairwise spatial and temporal relations for action recognition. In *ECCV'10: Proceedings of the Eighth European Conference on Computer Vision*, pages 508–521, 2010.

[127] M.A. Mendoza and N. Pérez de la Blanca. Hmm-based action recognition using contour histograms. In *IBPRIA'07: Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis, Part I*, pages 394–401, Girona, Spain, June 2007.

[128] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *ECCV'04: Proceedings of the Eighth European Conference on Computer Vision*, pages 69–82, 2004.

[129] T.B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 8(3):231–268, 2006.

[130] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 23(4):349–361, 2001.

[131] G. Mori and J. Malik. Estimating body configurations using shape context matching. In *ECCV'02: Proceedings of the IEEE European Conference on Computer Vision*, 2002.

[132] G. Mori, X. Ren, A.A. Efros, and J. Malik. Recovering human body configurationa. In *CVPR'04: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[133] F. Nater, H. Grabner, and L.V. Gool. Exploiting simple hierarchies for unsupervised human behavior analysis. In *CVPR'10: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[134] J.C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, 2008.

[135] H. Ning, W. Xu, Y. Gong, and T.S. Huang. Latent pose estimator for continuous action recognition. In *ECCV'08: European Conference on Computer Vision*, pages 419–433, 2008.

[136] S. Niyogi and E. Adelson. Analyzing and recognizing walking figures in xyt. In *CVPR'94: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 469–474, 1994.

[137] A.S Ogale, A. Karapurkar, G. Guerra-Filho, and Y. Aloimonos. View-invariant identification of pose sequences for action recognition. In *VACE'04: Proceedings of the VACE*, 2004.

[138] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, C.-C. Chen, Jong Taek Lee, Saurajit Mukherjee, J. K. Aggarwal, Hyungtae Lee, Larry Davis, Eran Swears, Xioyang Wang, Qiang Ji, Kishore Reddy, Mubarak Shah, Carl Vondrick, Hamed Pirsiavash, Deva Ramanan, Jenny Yuen, Antonio Torralba, Bi Song, Anesco Fong, Amit Roy-Chowdhury, and Mita Desai. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR'11: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[139] A. Oikonomopoulos, I. Patras, and M. Pantic. Spatiotemporal salient points for visual recognition of human actions. *IEEE Transactions on Systems, Man and Cybernetics* -, 36(3):710–719, 2006.

[140] B. Ommer and J. Malik. Multi-scale object detection by clustering lines. In *ICCV'09: Proceedings of the International Conference on Computer Vision*, 2011.

[141] A. Opelt, A. Pinz, and A. Zisserman. Learning an alphabet of shape and appearence for multi-class object detection. *International Jounral of Computer Vision*, 80(1):16–44, 2009.

[142] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4):807–832, 2002.

[143] S. Park and J.K. Aggarwal. Semantic-level understanding of human actions and interactions using event hierarchy. In *CVPRWS'04: Proceedings of the IEEE Computer Society Computer Vision and Pattern Recognition Workshop on Articulated and Non-Rigid Motion*, page 12, 2004.

[144] S. Pehlivan and P. Duygulu. A new pose-based representation for recognizing actions from multiple cameras. *Computer Vision and Image Understanding*, 115:140–151, 2011.

[145] M. Pierobon, M. Marcon, A. Sarti, and S. Tubaro. 3-d body posture tracking for human action template matching. In *ICASSP'06: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2006.

[146] R. Polana and R. Nelson. Low level recognition of human motion. In *NAM'94: Proceedings of the IEEE Workshop on Non-rigid and Articulated Motion*, pages 129–134, 1992.

[147] R. Polana and R. Nelson. Recognition of motion from temporal texture. In *CVPR'92: Proceedings of the IEEE Computer Society Conference on Computer Vision*, pages 129–134, 1992.

[148] R. Poppe. A survey on vision-based human action recognition. *Image Vision Computing*, 28(6):976–990, 2010.

[149] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Institute of Electrical and Electronics Engineers*, 2:257–286, 1989.

[150] H. Ragheb, S.A. Velastin, P. Remagnino, and T. Ellis. Human action recognition using robust power spectrum features. In *ICIP'08: Proceedings of the International Conference on Image Processing*, pages 753–756, 2008.

[151] D. Ramanan, D. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 29(1):65–81, 2007.

[152] K. Reddy, J. Liu, and M. Shah. Incremental action recognition using feature tree. In *ICCV'09: Proceedings of the IEEE 12th International Conference on Computer Vision*, 2009.

[153] J. Rittscher and A. Blake. Classification of human body motion. In *ICCV'99: Proceedings of the IEEE International Conference on Computer Vision*, pages 634–639, 1999.

[154] S. Sadek, A. Al-Hamadi, B. Michaelis, and U. Sayed. Toward robust action retrieval in video. In *BMVC'10: Proceedings of the British Machine Vision Conference*, 2010.

[155] B. Saghafi, E. Farahzadeh, D. Rajan, and A. Sluzek. Embedding visual words into concept space for action and scene recognition. In *BMVC'10: Proceedings of the British Machine Vision Conference*, 2010.

[156] H. Samet and M. Tamminen. Efficient component labeling of images of arbitrary dimension represented by linear bin-trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):579–586, 1988.

[157] K. Schindler and L. van Gool. Action snippets: How many frames does human action recognition require. In *CVPR'08: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[158] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[159] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR'04: Proceedings of the International Conference on Pattern Recognition*, pages 32–36, Cambridge, UK, August 2004.

[160] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ICM'07: Proceeding of the ACM International Conference on Multimedia*, 2007.

[161] L. Shao and R. Gao. A wavelet based local descriptor for human action recognition. In *BMVC'10: Proceedings of the British Machine Vision Conference*, 2010.

[162] Y. Sheikh, M. Sheikh, and M. Shah. Exploring the space of a human action. In *ICCV'05: Proceedings of the IEEE International Conference on Computer Vision*, pages 144–149, 2005.

[163] V.D. Shet, J. Neumann, V. Ramesh, and L.S. Davis. Bilattice based logical reasoning for human detection. In *CVPR'07: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, USA, 2007.

[164] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR'11: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

[165] C. Siagian and L. Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):300–312, Feb 2007.

[166] C. Siagian and L. Itti. Comparison of gist models in rapid scene categorization tasks. In *VSS'08: Proceedings of the Vision Science Society Annual Meeting*, May 2008.

[167] L. Sigal and M.J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1-2):4–27, 2010.

[168] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV'03: Proceedings of the International Conference on Computer Vision*, pages 1470–1477, 2003.

[169] J. Sivic and A. Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):591–606, 2009.

[170] N. Slonim and N. Tishby. Agglomerative information bottleneck. In *NIPS'99: Proceedings of the Neural Information Processing Systems Foundation*, 1999.

[171] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. In *CVPR'05: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 390–397, 2005.

[172] R. Souvenir and J. Babbs. Learning the viewpoint manifold for action recognition. In *CVPR'08: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[173] J. Starck and A. Hilton. Surface capture for performance based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31, 2007.

[174] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR'99: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2246–2252, 1999.

[175] J. Sun, Y. Mu, S. Yan, and L-F. Cheong. Activity recognition using dense long-duration trajectories. In *ICME'10: IEEE International Conference on Multimedia and Expo*, pages 322–327, 2010.

[176] X. Sun, M. Chen, and A. Hauptmann. Action recognition via local descriptors and holistic features. In *CVPR'09: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.

[177] A. Sundaresan, A. RoyChowdhury, and R. Chellappa. A hidden markov model based framework for recognition of humans from gait sequences. In *ICIP'03: Proceedings of the International Conference on Image Processing*, pages 93–96, Barcelona, Catalunia, Spain, September 2003.

[178] A. Swadzba, N. Beuter, J. Schmidt, and G. Sagerer. Tracking objects in 6d for reconstructing static scenes. In *CVPR Workshops*, 2008.

[179] C. Thurau and V. Hlavac. Pose primitive based human action recognition in videos or still images. In *CVPR'08: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[180] D. Tran and A. Sorokin. Human activity recognition with metric learning. In *ECCV'08: Proceedings of the European Conference on Computer Vision*, 2008.

[181] P. Turaga, R. Chellappa, V.S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, 9 2008.

[182] P. Turcot and D.G. Lowe. Better matching with fewer features: The selection of useful features in large database recognition problems. In *ICCV Workshops'09: Proceeding of the International Conference on Computer Vision Workshop*, 2009.

[183] H. Uemura, S. Ishikawa, and K. Mikolajczyk. Feature tracking and motion compensation for action recognition. In *BMVC'08: Proceedings of the British Machine Vision Conference*, 2008.

[184] M.M. Ullah, S.N. Parizi, and I. Laptev. Improving bag-of-features action recognition with non-local cues. In *BMVC'10: Proceedings of the British Machine Vision Conference*, 2010.

[185] V. Vapnik. Estimation of dependences based on empirical data [in russian]. Nauka, Moscow, 1979.

[186] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(3):475–480, 2005.

[187] P.A. Viola and M.J. Jones. Robust real-time face detection. In *ICCV'01: Proceedings of the IEEE Conference on Computer Vision*, pages 747–754, 2001.

[188] S.N. Vitaladevuni, V. Kellokumpu, and L.S. Davis. Action recognition using ballistic dynamics. In *CVPR'08: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[189] H. Wang, A. Klaser, C. Schmid, and C-L Liu. Action recognition by dense trajectories. In *CVPR'11: Proceedings of the IEEE Conference on Computer Vision*, pages 3169–3176, 2011.

[190] H. Wang, M.M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC'09: Proceedings of the British Machine Vision Conference*, page 127, September 2009.

[191] L. Wang and D. Suter. Informative shape representations for human action recognition. In *ICPR'06: Proceedings of the International Conference on Pattern Recognition*, pages 1266–1269, Washington, DC, USA, 2006. IEEE Computer Society.

[192] S. Wang, K. Huang, and T. Tan. Human action recognition with pose similarity. In *CCPR'10: Chinese Conference on Pattern Recognition*, pages 1–5, 2010.

[193] Y. Wang, K. Huang, and T. Tan. Human activity recognition based on r transform. In *CVPR'07: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 18–23, 2007.

[194] Y Wang and G. Mori. Hidden part models for human action recognition: Probabilistic versus max margin. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 33(7):1310–1323, 2011.

[195] Y. Wang and Greg Mori. Learning a discriminative hidden part model for human action recognition. In *NIPS'08: Proceedings of the Neural Information Processing Systems Foundation*, pages 1721–1728, 2008.

[196] J.A. Webb and J.K. Aggarwal. Structure from motion of rigid and jointed objects. *Artif. Intell.*, 19(1):107–130, 1982.

[197] D. Weinland and E. Boyer. Action recognition using exemplar-based embedding. In *CVPR'08: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–7, Anchorage, États-Unis, 2008.

[198] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. In *ICCV'07: Proceedings of the International Conference on Computer Vision*, pages 1–7, Rio de Janeiro, Brazil, 2007.

[199] D. Weinland, M. Özuysal, and P. Fua. Making action recognition robust to occlusions and viewpoint changes. In *ECCV'10: Proceedings of the European Conference on Computer Vision*, 2010.

[200] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2):249–257, 2006.

[201] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, 2011.

[202] G. Willems, T. Tuytelaars, and L.V. Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV'08: Proceedings of the European Conference on Computer Vision*, 2008.

[203] S.F. Wong and R. Cipolla. Extracting spatiotemporal interest points using global information. In *ICCV'07: Proceedings of the 11th IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[204] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):274–266, November 2007.

[205] X. Wu, W. Liang, and Y. Jia. Incremental discriminative-analysis of canonical correlations for action recognition. In *ICCV'09: Proceedings of the IEEE 12th International Conference on Computer Vision*, 2009.

[206] J. Yamato, J. Ohya, and K.Ishii. Recognizing human action in time-sequential images using hidden markov model. In *CVPR'92: Proceedings of the Eleventh IEEE International Conference on Computer Vision*, pages 379–385, 1992.

[207] P. Yan, S.M. Khan, and M. Shah. Learning 4d action feature models for arbitrary view action recognition. In *CVPR'08: Proceedings of the IEEE Computer Society Conference of Computer Vision*, 2008.

[208] C. Yang, Y. Guo, H. Sawhney, and R. Kumar. Learning actions using robust string kernels. In *HMLNCS'07:Proceedings of the 2nd Conference on Human motion: understanding, modeling, capture and animation*, pages 313–327, Berlin, Heidelberg, 2007. Springer-Verlag.

[209] M. Yang, F. Lv, W. Xu, K. Yu, and Y. Gong. Human action detection by boosting efficient motion features. In *ICCV'09: Proceedings of the IEEE International Conference on Computer Vision*, 2009.

[210] W. Yang, Y. Wang, and G. Mori. Recognizing human actions from still images with latent poses. In *CVPR'08: Proceedings of the IEEE Computer Society Conference on Computer Vision*, 2010.

[211] A. Yao, J. Gall, and L.V. Gool. A hough transform-based voting framework for action recognition. In *CVPR'10: Proceedings of the Eleventh IEEE International Conference on Computer Vision*, 2010.

[212] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. In *CVPR'05: Proceedings of the IEEE Computer Society Conference on Computer Vision*, 2005.

[213] A. Yilmaz and M. Shah. Recognizing human actions in videos acquired by uncalibrated moving cameras. In *ICCV'05: Proceedings of the IEEE International Conference on Computer Vision*, volume 1, pages 150–157, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

[214] S.M. Yoon and A. Kuijper. Human action recognition using segmented skeletal features. In *ICPR'10: International Conference on Pattern Recognition*, pages 3740–3743, Istanbul, Turkey, August 2010.

[215] G. Yu, J. Yuan, and Z. Liu. Unsupervised random forest indexing for fast action search. In *CVPR'11: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 865–872, 2011.

[216] T. Yu, T. Kim, and R. Cipolla. Real-time action recognition by spatio-temporal semantic and structural forests. In *BMVC'10: Proceedings of the British Machine Vision Conference*, 2010.

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

[218] L. Zelnik-Manor and M. Irani. Event-based analysis in video. In *CVPR'01: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 123–130, 2001.

[219] Z. Zhang, Y. Hu, S. Chan, and L-T. Chia. Motion context: A new representation for human action recognition. In *ECCV'08: Proceedings of the European Conference on Computer Vision*, pages 817–829, 2008.