ISTA - ISCTE-IUL School of Technology and Architecture

Human Activity Recognition and Prediction in RGB-D Videos

David Walter Figueira Jardim

Thesis specially presented for the fulfillment of the degree of
Doctor in Information Science and Technology

Supervisor
Dr. Luís Miguel Martins Nunes, Assistant Professor
ISCTE-Instituto Universitário de Lisboa
Co-Supervisor
Dr. Miguel Sales Dias, Assistant Professor
ISCTE-Instituto Universitário de Lisboa

February 2018

ISTA - ISCTE-IUL School of Technology and Architecture

Human Activity Recognition and Prediction in RGB-D Videos

David Walter Figueira Jardim

Thesis specially presented for the fulfillment of the degree of
Doctor in Information Science and Technology

Jury:
Dr. Manuel João da Fonseca, Associate Professor, Faculdade de Ciências da Universidade de Lisboa
Dr. Rosaldo Rosetti, Assistant Professor, Faculdade de Engenharia da Universidade do Porto
Dr. Artur Ferreira, Assistant Professor, Instituto Superior de Engenharia de Lisboa
Dr. Pedro Faria Lopes, Associate Professor, Instituto Universitário de Lisboa
Dr. Bráulio A. Barreira Alturas, Assistant Professor, Instituto Universitário de Lisboa
Dr. Luís Miguel Martins Nunes, Assistant Professor, Instituto Universitário de Lisboa

February 2018

Resumo

Reconhecimento de atividade humana é uma área de investigação multidisciplinar que tem atraído o interesse de investigadores especializados em aprendizagem automática, visão por computador e medicina. Esta área tem diversas aplicações: sistemas de vigilância, interação homem-máquina, análise de desportos, robôs colaborativos, saúde e automóveis autónomos. Capturar atividade humana apresenta dificuldades técnicas como oclusão, iluminação insuficiente, seguimento erróneo e questões éticas. O movimento humano pode ser ambíguo e com múltiplas intenções. A forma como interagimos com outros seres humanos e objetos cria uma combinação quase infinita de variações de como fazemos as coisas. O objetivo desta dissertação é desenvolver um sistema capaz de reconhecer e prever a atividade humana usando técnicas de aprendizagem automática para extrair significado de características calculadas a partir de articulações do corpo humano capturado pela câmara Kinect. Propomos uma arquitetura hierárquica e modular que realiza segmentação temporal de sequências de ações, anotação semi-supervisionada de sub-atividades utilizando técnicas de *clustering*, reconhecimento de sub-atividade *frame-a-frame* em tempo real usando classificadores binários de *random decision forests* logo a partir dos primeiros instantes da ação e previsão de atividade em tempo real baseada em *conditional random fields* para modelar a estrutura das sequências de ações para obter as futuras possibilidades.

Gravámos um novo conjunto de dados contendo sequências de ações agressivas com um total de 72 sequências, 360 amostras de 8 ações distintas realizadas por 12 sujeitos. Efetuamos testes extensivos com dois conjuntos de dados, comparando o desempenho de reconhecimento de vários classificadores supervisionados treinados com dados anotados manualmente ou com dados anotados de forma semi-supervisionada. Aprendemos como a qualidade dos conjuntos de treino afeta os resultados que dependem também da complexidade das ações que estão a ser reconhecidas. Conseguímos obter melhores resultados que algumas das abordagens existentes na literatura em reconhecimento de atividade, efetuamos o reconhecimento de forma antecipada e obtivemos resultados encorajadores na previsão de atividades.

Palavras-chave: Kinect, RGB-D, deteção de articulações, segmentação temporal, anotação, análise movimento humano, reconhecimento de ações, previsão de ações, antecipação, aprendizagem automática.

Abstract

Human Activity Recognition is an interdisciplinary research area that has been attracting interest from several research communities specialized in machine learning, computer vision, and medical research. The potential applications range from surveillance systems, human computer interfaces, sports analysis, digital assistants, collaborative robots, health-care and self-driving cars. Capturing human activity presents technical difficulties like occlusion, insufficient lighting, unreliable tracking and ethical concerns. Human motion can be ambiguous and have multiple intents. The complexity of our lives and how we interact with other humans and objects prompt to a nearly infinite combination of variations in how we do things.

The focus of this dissertation is to develop a system capable of recognizing and predicting human activity using machine learning techniques to extract meaning from features computed from relevant joints of the human body captured by the skeleton tracker of the Kinect sensor. We propose a modular framework that performs off-line temporal segmentation of sequences of actions, off-line semi-unsupervised labeling of sub-activities via clustering techniques, real-time frame-by-frame sub-activity recognition using random decision forest binary classifiers right from the very first frames of the action and real-time activity prediction with conditional random fields to model the sequential structure of sequences of actions to reason about future possibilities. We recorded a new dataset containing long sequences of aggressive actions with a total of 72 sequences, 360 samples of 8 distinct actions performed by 12 subjects. We experimented extensively with two different datasets, compared the recognition performance of several supervised classifiers trained with manually labeled data versus semi-unsupervised labeled data. We learned how the quality of the training data affects the results which also depends on the complexity of the actions being recognized. We outperformed state-of-the-art activity recognition approaches, performed early action recognition and obtained encouraging results in activity prediction.

Keywords: Kinect, RGB-D, skeletal-tracking, temporal segmentation, labeling, human motion analysis, action recognition, action prediction, anticipation, machine learning.

Acknowledgements

I am very grateful to my supervisors Prof. Dr. Luís Miguel Martins Nunes and Prof. Dr. Miguel Sales Dias. Prof. Dr. Luís Miguel Martins Nunes was relentless on his efforts to provide support and guidance all the way through this roller-coaster which is a Ph.D, his insightful contributions were decisive on my success. I would like to thank Prof. Dr. Miguel Sales Dias for the opportunity to pursue a Ph.D at Microsoft Portugal. I appreciate the vote of confidence and all the support that he has provided, the ongoing discussions in meetings helped me to stimulate my research and development.

I thank all my friends, who have encouraged and helped me, particularly Miguel Duarte, Jairo Avelar, André Santos, Ricardo Carvalho and all my colleagues at Microsoft Language Development Center who contributed to the recording of the PRECOG dataset.

I gratefully acknowledge the love and support of all my family, specially my parents, Álvaro and Fátima Figueira, my sisters and my brother. I would also like to thank my uncles, that always treated me as their own son. Unfortunately my uncle passed away before seeing me graduate, but I know that he is somewhere seeing it! Finally, my amazing girlfriend Joana, for her love, patience, friendship, support, and encouragement which everyday leads me to being a better human being.

This work has been supported by the Portuguese Foundation for Science and Technology (Fundação para a Ciência e Tecnologia) under the grant SFRH/BDE/52125/2013 and by Microsoft Portugal.

Contents

List of Figures

List of Algorithms

List of Tables

Acronyms

CAD	Cornell Activity Dataset
CFG	Context-Free-Grammars
CRF	Conditional Random Fields
CSV	Comma Separated Value
CV	Computer Vision
DBN	Dynamic Bayesian Networks
HAP	Human Activity Prediction
HAR	Human Activity Recognition
HMDP	Hidden variable Markov Decision Process
HMM	Hidden Markov Models
HOD	Histogram of Oriented Displacements
MDM	Multiple Dynamic Models
MDP	Markov Decision Process
ML	Machine Learning
MLP	Multilayer Perceptron
MMTC	Maximum Margin Temporal Clustering
NLP	Natural Language Processing
POS	Part-of-Speech Tagging
PRECOG	Prediction and Recognition Framework
RF	Random Decision Forests
RGB-D	Red Green Blue - Depth
SCFG	Stochastic Context-Free-Grammars
ST	Skeletal Tracking
SVM	Support Vector Machines
WKM	Warped K-means

Dedicated to my family

Chapter 1

Introduction

"Intellectual growth should commence at birth and cease only at death."

Albert Einstein

Understanding human behavior in intricate real-life scenarios is one of the greatest challenges in the areas of computer vision and machine learning. It is a very complex task which encompasses multiple aspects: from recognizing the activities that humans are performing, to recognizing the objects present in the environment and how humans interact with the recognized objects or with other humans. The ability to understand human behavior would foster the creation of new applications and benefit existing ones. Examples include video surveillance systems, human-computer interaction, robotics for human collaboration and autonomous vehicles.

Most of the previous research in human activity recognition has been focused in 2D images and videos, with emphasis on recognizing human poses and objects. Recent advances in Red Green Blue - Depth (RGB-D) sensors have encouraged the development of next-gen applications that attempt to solve complex problems. RGB-D sensors provide access to 3D depth information, amongst other features. This extra dimension has enabled researchers to obtain accurate 3D structures of the scenes, objects and human poses, thus significantly improving computer

vision solutions. This development in technology has spawned several sub-fields of research that focus on 3D object recognition, 3D scene understanding and 3D action recognition. The relative low price of RGB-D has made them accessible to consumers and researchers, leading to an increase of visual data collected about people performing actions, to a point where the most recent dataset has 56 thousand video samples and 4 million frames (Shahroudy et al., 2016). However, even with recent advances, a system that performs real-time generic action recognition and prediction is still a rare thing.

Human Activity Recognition (HAR) and Human Activity Prediction (HAP) is very challenging (Aggarwal and Ryoo, 2011; Ryoo, 2011; Kong and Fu, 2015). Aside from image acquisition problems like background clutter, partial occlusion, changes in scale, viewpoint and lighting, human motion analysis is difficult for several reasons. The way in which each individual executes a given action will vary because of their inherent anatomy and habits, giving different expressions to actions within the same class. On the other hand, actions from different classes might be difficult to differentiate if they are very similar. Actions can take place over long periods of time, with infinite combinations of sub-activities. Humans can perform different actions to achieve the same goal. These nuances and ambiguities make the process of extracting meaningful information from data a very arduous task.

Due to the acceleration of technology, we now live in a data-driven era. This has enabled the application of multiple Machine Learning (ML) methods which require large amounts of labeled data to learn upon. The application of these ML methods to HAR require large datasets of human activity. Annotating large datasets of human behavior is time consuming, error prone and requires knowledge of the specific event. Finally, HAR must be done in real-time with a response time that will allow the system to act upon the recognized activity, sometimes before its completion.

1.1 Problem Statement

The focus of this dissertation is to address some of the key challenges in the field of HAR through the proposal and study of a hierarchical model designed to represent a human motion understanding system (Prediction and Recognition Framework (PRECOG)) with several layers of abstraction, from low-level data acquisition to high-level action prediction. The approach consists of decomposing an activity (sequence of actions) into several recognizable and predictable actions with ML techniques that use features computed from the estimated position of the skeleton joints in 3D, provided by Microsoft's Kinect sensor.

This approach was presented to address the following challenges in the field of HAR: (i) the ever increasing necessity of large amounts of labeled data; (ii) recognize generic human actions in real-time, before their completion; (iii) combine real-time action recognition and prediction into a system capable of understanding and foreseeing human motion from observable patterns. Figure 1.1 illustrates how different ML techniques i.e., K-means clustering, Random Decision Forests (RF), and Conditional Random Fields (CRF) are combined to perform temporal segmentation, clustering, detection and classification of human postures and finally given the sequential nature of high-level activities, anticipate the next action that the human subject will perform.

We evaluate our approach with two datasets which contemplate several different scenarios. The approach is first validated in the PRECOG dataset (recorded by us) which contains long sequences of aggressive actions. Then, we experimented with the Cornell Activity Dataset (CAD)-120 dataset which has sequences of long daily activities. This allowed us to submit our approach to a variety of situations evaluating the system's robustness. We also validate our approach by conducting real-time experiments. Our approach presents a solution to several of the problems described above and advances the understanding of human motion a step closer to the understanding that humans perform, continuously, effortlessly and quite accurately.

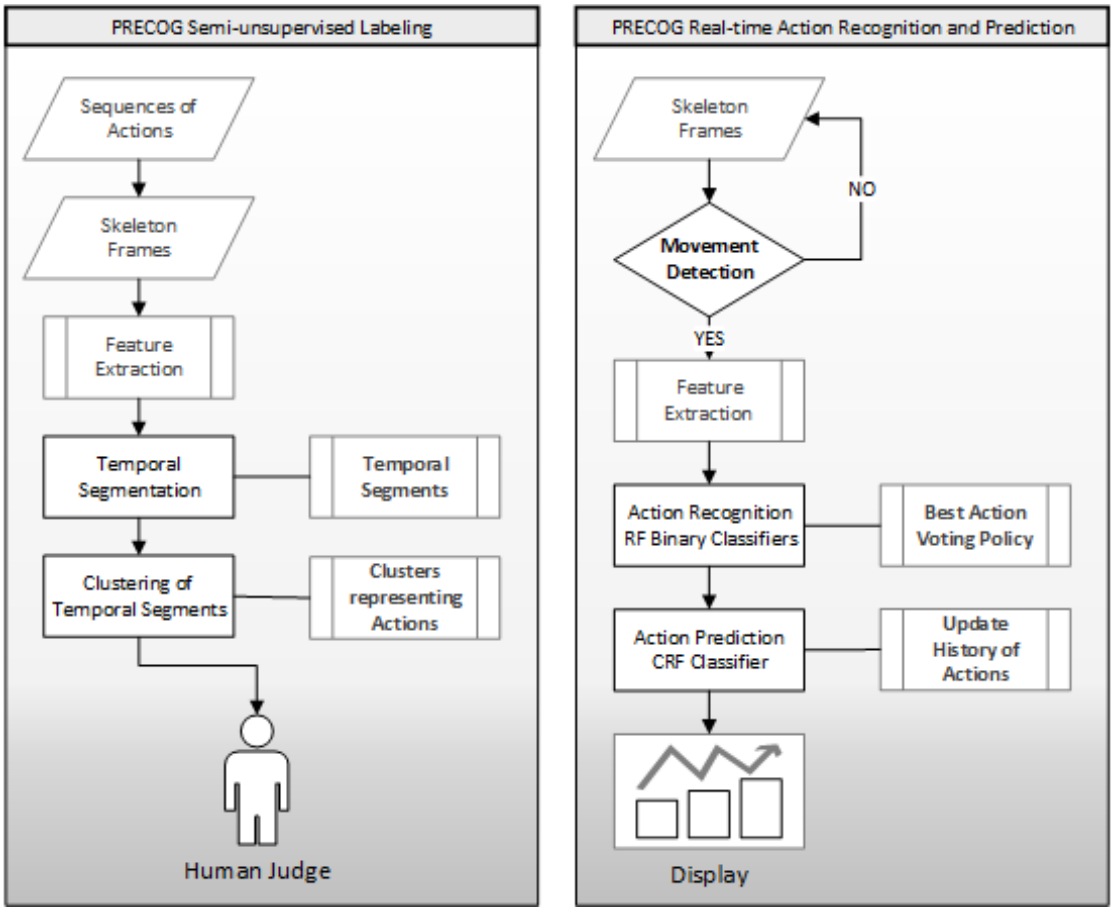

FIGURE 1.1: The system architecture of PRECOG comprising two main sections. (a) Workflow of the semi-unsupervised labeling process responsible for labeling the human activity data and train the classifiers (*OFFLINE*). (b) Workflow of the real-time action recognition and prediction process (*ONLINE*).

1.2 Objectives

The aim of this dissertation will be accomplished by fulfilling the following research objectives:

- Capture, label and publish a dataset of human activity containing full sequences of actions in RGB-D data.
- Reduce the amount of input required by a human judge to label a human activity dataset by proposing a method to perform semi-unsupervised labeling of data.
- Measure the impact of the noise introduced by semi-unsupervised labeling of human activity used for training.

- Perform real-time action recognition of complex human activities using ML algorithms.
- Model the hierarchical structure of sequences of actions to perform real-time action prediction of the next action that will occur.

1.3 Contribution of Research

The contributions of this thesis are the following: 1) A newly recorded and published dataset for RGB-D human action recognition, which contains 72 videos of sequences of activities collected from 12 subjects, instead of isolated actions. At the moment of its recording we did not find datasets with long sequences of actions publicly available; 2) We present and compare several methods for performing temporal segmentation of sequences of actions; 3) We demonstrate that semi-unsupervised labeling of RGB-D videos containing human activity is possible depending on the complexity/similarity of the actions; 4) We compare the performance of our framework trained with manually labeled data versus semi-unsupervised labeled data; 5) We implement and demonstrate a framework capable of recognizing human activity in real-time and perform early recognition, often based only on the initial frames of the action, obtaining results that surpass the state-of-the art for similar datasets, including parallel work developed during the span of this thesis (Koppula et al., 2013; Nirjon et al., 2014; Gaglio et al., 2015; Cippitelli et al., 2016); 6) We demonstrate that CRF can be used to anticipate the next possible action performed by a subject based on the history of actions executed.

Below we present the scientific publications that resulted from our research:

- D. Jardim, L. Nunes and M. S. Dias, "**Human activity recognition and prediction**", *in Proceedings of the Doctoral Consortium in The International Conference on Pattern Recognition Applications (ICPRAM)*, SCITEPRESS Digital Library, 2015, pp. 24–32.

- D. Jardim, L. Nunes and M. S. Dias, "**Automatic human activity segmentation and labeling in rgb-d videos**", *in Proceedings of the 8th International KES Conference on Inteligent Decision Technologies (KES-IDT)*, Springer International Publishing, 2016, p. 383.

- D. Jardim, L. Nunes and M. S. Dias, "**Impact of automated action labeling in classification of human actions in RBG-D videos**", *in Proceedings of the 22nd European Conference in Artificial Intelligence (ECAI)*, IOS Press, 2016, p. 1632.

- D. Jardim, L. Nunes and M. S. Dias, "**Human activity recognition from automatically labeled data in RGB-D videos**", *in Proceedings of the 8th Computer Science and Electronic Engineering Conference (CEEC)*, IEEE Press, 2016, p. 89.

- D. Jardim, L. Nunes and M. S. Dias, "**Predicting human activities in sequences of actions in RGB-D videos**", *in Proceedings of 9th International Conference on Machine Vision (ICMV 2016)*, SPIE Digital Library, 2017. **Best Presentation Award**

1.4 Thesis Structure

In this section, we provide an overview of the thesis structure and the main sections of each chapter. In Chapter 2, we review the current state of the art in the field of human activity recognition and prediction. The chapter is organized in four major sections: a review of the current state of the art in the field of human motion analysis with different activity recognition methodologies, an overview of the HAP field, a brief discussion of the publicly available datasets, and how 3D vision was revolutionized thanks to the availability of low-cost and light-weight RGB-D sensors such as the Microsoft Kinect sensor.

In Chapter 3, we present the hierarchical model describing our proposal for action recognition and prediction. In our methodology, the hierarchical model spans

through four levels of abstraction with a *bottom-up* approach, ranging from low-level data acquisition from the sensor, feature extraction and engineering, low-level action recognition to high-level action prediction (see Section 3.1). We propose a method to perform semi-unsupervised labeling of human activity in RGB-D videos with a combination of temporal segmentation and clustering methods with the purpose of reducing the amount of input required from a human judge to label a human activity dataset. Finally, we described our own recorded and labeled PRECOG dataset which contains sequences of aggressive actions.

In Chapter 4 we validate our temporal segmentation and clustering methods for performing semi-unsupervised labeling of human activity in RGB-D videos (see Section 4.3). We demonstrate two different methods to perform real-time human activity recognition using machine learning classifiers with results that outperform state of the art approaches. We explore the importance of having data accurately labeled with a series of experiments. These experiments compare the performance of the recognition classifiers trained with semi-unsupervised labeled data versus manually labeled data and present the results (see Section 4.4).

In Chapter 5 we implement our full hierarchical model to accomplish action recognition and prediction. We demonstrate the ability of our approach to perform early activity recognition. From the conducted experiments the system has proven itself capable of classifying the correct action with only a few frames of the action (see Section 5.2). An additional set of experiments is conducted to validate our two approaches to perform activity prediction: n-gram action prediction and CRF action prediction. Again we perform experiments to compare the performance of the classifiers trained with data generated from the semi-unsupervised labeled data versus data manually labeled (see Sections 5.3, 5.4).

In Chapter 6, we conclude the thesis and discuss future directions of research. The final results are compiled in a journal article that is currently under evaluation.

Chapter 2

State of the Art

"An investment in knowledge pays the best interest."

Benjamin Franklin

Human activity recognition is a computer vision problem in which activities performed by humans from video data are automatically recognized. HAR is an important and challenging task in the area of computer vision research (Aggarwal and Ryoo, 2011). The goal of HAR is to analyze and detect ongoing activities from video. Although significant progress has been made, recognizing human activities from video sequences or still images is a challenging task due to problems, such as background clutter, partial occlusion, changes in scale, viewpoint, lighting, and appearance (Vrigkas et al., 2015). Most of the existing systems designed to identify specific activities in a live feed or search in video archives still rely on human resources. Manual analysis of video is labor intensive, fatiguing, and error prone. Solving the problem of recognizing human activities from video can lead to improvements in several application fields (Figure 2.1) like surveillance systems, human computer interfaces, sports video analysis, digital shopping assistants, video retrieval, gaming and health care (Niu et al., 2004; Intille and Bobick, 1999; Nirjon et al., 2014; Popa et al., 2012; Keller et al., 2011).

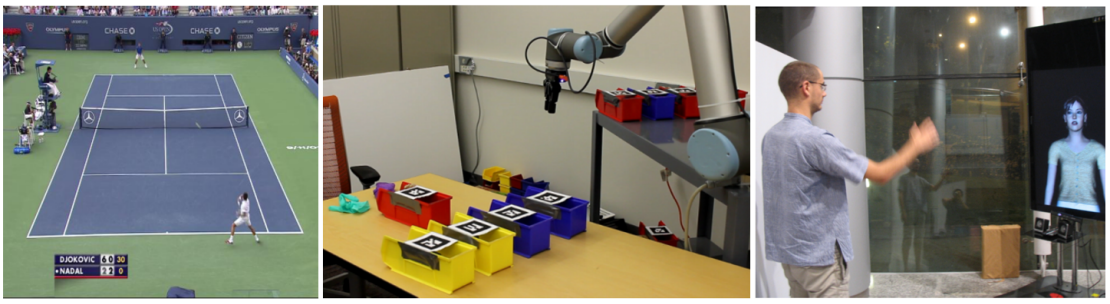

FIGURE 2.1: Multiple applications for action recognition, sports, human-robot collaborative scenarios and digital assistants.

Source: Li et al. (2012); Hawkins et al. (2013); Elgendi et al. (2012)

Some of the earliest work on extracting useful information through video analysis was performed by J. O'Rourke and Badler (1980) in which images were fitted to an explicit constraint model of human motion, with constraints on human joint motion, and constraints based on the imaging process. Also Rashid (1980) did some work on understanding the motion of 2D points in which he was able to infer 3D position. Driven by application demands, this field has seen a relevant growth in the past decade with multiple applications such as surveillance systems, human computer interfaces, video retrieval, gaming and quality-of-life devices for the elderly (Niu et al., 2004; Intille and Bobick, 1999; Nirjon et al., 2014; Popa et al., 2012; Keller et al., 2011).

Most of the activity recognition research focused only on recognizing the actions after they occurred and were not suited to perform early recognition of unfinished actions. In real world scenarios it should be more useful to have the ability to predict the current ongoing activity before the activity is fully executed. This fostered a new field of research called Human activity prediction. First steps were given through early detection on simple actions Ryoo (2011) to more recently the ability to perform activity anticipation with a few seconds ahead in time Koppula and Saxena (2016). Below, we provide an overview of the fields of Human activity recognition and prediction, the existing datasets and the sensors used to record them.

2.1 Human Activity Recognition

Initially the main focus of human activity recognition was to recognize simple human actions such as walking and running (Gavrila, 1999). Now, that problem is well explored and researchers are moving towards recognition of complex realistic human activities involving multiple persons and objects. In the review written by Aggarwal and Ryoo (2011) an approach-based taxonomy was chosen to categorize the activity recognition methodologies which are divided into three main categories: statistical approaches, syntactic approaches and description based approaches. Since then this categorization was adopted in other surveys by Vishwakarma and Agrawal (2013) and Onofri et al. (2016).

Single-layered approaches (Bobick and Wilson, 1997; Yamato et al., 1992; Starner et al., 1998) typically represent and recognize human activities directly based on sequences of images and are suited for the recognition of gestures and actions with sequential characteristics. Hierarchical approaches represent high-level human activities that are composed of other simpler activities (Aggarwal and Ryoo, 2011). Since we are interested in recognizing high-level human activities we will focus on the hierarchical approaches. Hierarchical approaches can be seen as statistical, syntactic and description-based (Damen and Hogg, 2009; Gupta et al., 2009; Intille and Bobick, 1999; Pinhanez and Bobick, 1998; Ryoo and Aggarwal, 2009; Yu and Aggarwal, 2006).

2.1.1 Statistical Approaches

This approach uses multiple layers of statistical state-based models (usually two) such as Hidden Markov Models (HMM) (Rabiner and Juang, 1986) and Dynamic Bayesian Networks (DBN) (Fox et al., 2009) to recognize activities with sequential structures. At the lower-layer, atomic actions are recognized from sequences of feature vectors which are converted to a sequence of atomic actions. Then the upper-layer treats this sequence of atomic actions as observations generated by the

lower-layer models. For each model, a probability of the model generating a sequence of observations is calculated to measure the likelihood of a match between the activity and the input image sequence. One of the most fundamental forms of the hierarchical statistical approach was presented by Oliver et al. (2002) using layered Hidden Markov models. In this approach, the bottom layer HMM recognize atomic actions of a single person by matching the models with the sequence of feature vectors extracted from videos. The upper layer HMM represent a high-level activity as a sequence of atomic actions. The authors Nguyen et al. (2005) have also constructed hierarchical HMM of two layers to recognize complex sequential activities. These approaches are especially suited to recognize sequential activities (Damen and Hogg, 2009; Yu and Aggarwal, 2006). With enough training data, statistical models are able to reliably recognize activities even with noisy inputs. The major limitation of statistical approaches is their inability to recognize activities with complex temporal structures, such as an activity composed of concurrent sub-events (Ivanov and Bobick, 2000).

2.1.2 Syntactic Approaches

Syntactic approaches model human activities as a string of symbols, where each symbol corresponds to an atomic-level action which has to be recognized first. Human activities are represented as a set of production rules generating a string of atomic actions, and they are recognized by adopting parsing techniques from the field of programming languages such as Context-Free-Grammars (CFG) and Stochastic Context-Free-Grammars (SCFG). A hierarchical approach to the recognition of high-level activities using SCFG was proposed by Ivanov and Bobick (2000) where they divided the framework into two layers: the lower layer used HMM for the recognition of simple actions, and the higher layer used stochastic parsing techniques for the recognition of high-level activities. The authors in Moore and Essa (2002) extended the work described by Ivanov and Bobick (2000) using SCFG for the recognition of activities, focusing on multi-tasked activities. They were able to recognize human activities happening in a blackjack card game,

such as "a dealer dealt a card to a player" with a high accuracy level. This approach also struggles to recognize concurrent activities. Syntactic approaches model a high-level activity as a string of atomic-level activities that compose them. The temporal ordering of these atomic-level activities has to be strictly sequential. Therefore, they tend to have difficulties when an unknown observation interferes with the system.

2.1.3 Description-based approaches

This approach is a recognition approach that explicitly maintains spatio-temporal structures of human activities. It represents a high-level human activity in terms of simpler activities as sub-events, describing their temporal, spatial and logical relationships. The recognition of the activity is performed by searching the sub-events satisfying the relations specified in its representation. In description-based approaches, a time interval is usually associated with an occurring sub-event to specify necessary temporal relationships among sub-events. Many researchers (Pinhanez and Bobick, 1998; Nevatia et al., 2003; Vu et al., 2003; Ryoo and Aggarwal, 2006) have adopted the temporal predicates specified by Allen (2013). These predicates are: before, meets, overlaps, during, starts, finishes and equals. Researchers Pinhanez and Bobick (1998) have created a system that recognizes the top-level activity by checking which sub-events have already occurred and which have not. They were able to recognize cooking activities in a kitchen environment such as "picking up a bowl". The atomic-level actions were manually labeled from the video in the experiments, and recognition was successful even when one of the atomic actions was not provided.

A description-based approach to analyze plays in American football was designed by Intille and Bobick (1999). Using simple temporal predicates (before and around), they have shown that complex human activities can be represented by listing the temporal constraints in a format similar to those of programming languages. This representation was done using three levels of hierarchy: atomic-level, individual-level and team-level activities. More recently Ryoo and Aggarwal (2009)

proposed a probabilistic extension to their framework that is able to compensate for the failures of its low-level components. Description-based approaches are fragile when their low-level components are noisy. This limitation has been overtaken by Ryoo and Aggarwal (2009), where they have used a logistic regression to model the probability distribution of an activity, and used it to detect the activity even when some of its sub-events have been misclassified. Human activities with complex temporal structures can be represented and recognized by description-based approaches which can successfully handle concurrent organized sub-events. The major drawback of description-based approaches is their inability to compensate for the failures of low-level components (e.g., gesture detection failure). This issue has been addressed in some recent work done by Gupta et al. (2009) and Ryoo and Aggarwal (2009) where they introduce a probabilistic semantic-level recognition to cope with imperfect lower-layers.

Hoai et al. (2011) use a supervised framework which provides a systematic algorithm for time series segmentation and action recognition. The recognition model was trained discriminatively using multi-class Support Vector Machines (SVM), while segmentation inference was done efficiently with dynamic programming, though the proposed method yielded encouraging results on standard datasets, its requirement for fully labeled data for training inevitably limits its applicability to small training sets with a small number of actions. Another approach by Shi et al. (2011) presents a discriminative semi-Markov model approach, and define a set of features over boundary frames, segments as well as neighboring segments. They efficiently solve the inference problem of simultaneously segmentation and recognition using a Viterbi-like dynamic programming algorithm. Hoai and De la Torre (2012) propose Maximum Margin Temporal Clustering (MMTC), a learning framework that simultaneously performs temporal segmentation and learns a multi-class SVM for separating temporal clusters. They divide time series into a set of disjoint segments such that each segment belongs to a cluster. Maximum Margin Temporal Clustering (MMTC) maximizes the cluster separability using the SVM score as the measure of separability. The results obtained overcame the state-of-the-art algorithms at the time. The authors in Zhou et al. (2013) pose

the problem of learning motion primitives (actions) as a temporal clustering task, and derive, bottom-up, an unsupervised hierarchical framework called hierarchical aligned cluster analysis (HACA). HACA finds a partition of a given multidimensional time series into m disjoint segments such that each segment belongs to one of k clusters representing an action. Using motion capture data HACA is able to achieve competitive detection performances (77%) for human actions in a completely unsupervised fashion.

There are some approaches which combine motion information and object properties (Ramirez-Amaro et al., 2015; Wachter and Asfour, 2015). In Ramirez-Amaro et al. (2015) the authors abstract the problem in two stages. First, by recognizing general motions such as moving, not moving or tool used. Second, by reasoning about more specific activities (Reach, Take, etc.) given the current context, i.e. using the identified motions and the objects of interest as input information. They've obtained an accuracy classification of 92%. Wachter and Asfour (2015) propose a two-level hierarchical action segmentation (HAS) approach that takes into account contact relations between human end effectors, the scene, and between objects in the scene, using 6D pose trajectories extracted from marker-based tracking system. This work shows that HAS allows the identification of meaningful segments in complex human demonstrations without over-segmentation and without omitting important demonstration key frames.

2.1.4 Human activity detection from RGB-D videos

Recently, with the availability of affordable RGB-D sensors, which capture RGB-D data and in some cases are capable of providing joint level information in a non-invasive way, allowed the developers to abstract away from Computer Vision (CV) techniques and use 3D points to model postures. A parallel study (Koppula et al., 2013) using the Kinect sensor considers the problem of extracting a descriptive labeling of the sequence of sub-activities being performed by a human, and more importantly, of their interactions with the objects in the form of associated affordances. The learning problem is formulated using a structural support vector

machine (SSVM) approach, where labelings over various alternate temporal segmentations are considered as latent variables. The method obtained an accuracy of 79.4% for affordance, 63.4% for sub-activity and 75.0% for high-level activity labeling.

In Hussein et al. (2013) the covariance matrix for skeleton joint locations over time is used as a discriminative descriptor for a sequence of actions. To encode the relationship between joint movement and time, multiple covariance matrices are deployed over sub-sequences in a hierarchical fashion. Their experiments show that using the covariance descriptor with an off-the-shelf classification algorithm one can obtain an accuracy of 90.53% in action recognition on multiple datasets.

In a parallel work Gowayyed et al. (2013) propose a descriptor for 2D trajectories: Histogram of Oriented Displacements (HOD). Each displacement in the trajectory votes with its length in a histogram of orientation angles. 3D trajectories are described by the HOD of their three projections. HOD is used to describe the 3D trajectories of body joints to recognize human actions. The descriptor is fixed-length, scale-invariant and speed-invariant. Experiments on several datasets show that this approach can achieve a classification accuracy of 91.26%.

The method developed by Jia et al. (2014) addressed an interesting problem of transferring depth information to a target of RGB action data (depth data is not available) and used both RGB data and the learned depth data for action recognition. By borrowing an auxiliary dataset, with both RGB and depth data they are capable of uncovering missing depth information in the target data, couple two modalities (RGB and depth) and capture structure information. From their experiments they achieved superior performance over existing methods with accuracy values of 92.09%.

More directly related to our research, Nirjon et al. (2014) developed a system called Kintense which is a real-time system for detecting aggressive actions from streaming 3D skeleton joint coordinates obtained from Kinect sensors. In two multi-person households it achieves up to 90.0% accuracy in action detection with a combination of supervised classifiers to recognize a set of predefined actions

helped by human feedback to reduce false alarms. In the following year Gaglio et al. (2015) presented another Kinect based approach which also uses the joints of the human body, combining three different machine learning techniques (K-means clustering, support vector machines, and hidden Markov models) to recognize the body postures while performing an activity, modeling each activity as a spatiotemporal evolution of known postures with an average accuracy of 92.5%. Recently Cippitelli et al. (2016) proposed an activity recognition algorithm also using skeleton data extracted by RGB-D sensors. They extract key poses from the skeleton to compose a feature vector. These key poses are associated using a clustering algorithm. They perform action recognition with the help of a multi-class support vector machine and managed to outperform some of the state-of-the-art results.

The activity recognition approach presented in this thesis differs from the previously discussed approaches in a number of key aspects: (i) depending on the complexity and ambiguity of the actions our approach does not require a fully annotated dataset to train the classifiers, it only requires a high-level class identification from a human judge; (ii) we tested our approach in long sequences of actions, instead of short isolated actions, this added an extra layer of complexity which required some experimentation with supervised and unsupervised temporal segmentation methods; (iii) our multi-class action recognition method is capable of real-time action recognition outperforming some of the some of the state-of-the-art results while offering early recognition right from the initial frames of the action being performed.

2.2 Human Activity Prediction

In our daily activities we perform prediction or anticipation when interacting with other humans or with objects. Prediction of human activity made by computers can be applied in surveillance systems (Ziebart et al., 2009), safety systems (Keller et al., 2011), autonomous vehicles and shopping assistances (Popa et al., 2012) (Figure 2.2).

FIGURE 2.2: Possible applications for activity prediction, autonomous vehicles, surveillance systems and human-robot collaborative scenarios.

Source: Waymo Alphabet (2017); TAV IT (2017); Nikolaidis et al. (2013)

HAP is a probabilistic process of inferring ongoing activities from videos (Ryoo, 2011). The problem of predicting unknown variables had a major breakthrough in 1961 with the work developed by Kalman and Bucy (1961) commonly known as the Kálmán filter. This algorithm works in a two-step process. In the prediction step, the Kálmán filter produces estimates of the current state variables, along with their uncertainties. Once the outcome of the next measurement (including random noise) is observed, these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. It has been applied in guidance, control of vehicles and time series analysis. The Kálmán filter can also be applied in HAP as we have seen in Pentland and Liu (1999); Ziebart et al. (2009). One of the earliest approaches that we have found tried to model and predict human behavior when driving an automobile from Pentland and Liu (1999). Their goal is to recognize human driving behaviors accurately and anticipate the human's behavior for several seconds into the future. They consider the human as a device with a large number of internal mental states, each with its own particular control behavior and interstate transition probabilities. The states of the model can be hierarchically organized to describe both short-term and longer-term behaviors; for instance, in the case of driving an automobile, the longer-term behaviors might be passing, following, and turning, while shorter-term behaviors would be maintaining lane position and releasing the brake. The authors introduced the concept of Multiple Dynamic Models (MDM) which defends that the most complex model of human behavior is to have several alternative models of the person's dynamics. Then, at each instant, they make observations of the person's state, decide which

model applies, and give a response based on that model. This multiple model approach produces a generalized maximum likelihood estimate of the current and future values of the state variables. With this approach they have accurately categorized human driving actions very soon after the beginning of the action. There are other recent works that also address the task of early recognition (Ryoo, 2011; Hoai and De La Torre, 2014).

Another type of prediction was addressed by Ziebart et al. (2009) where a robot should predict the future locations of people and plan routes that will avoid disrupting the person's natural behavior due to the robot's proximity, while still efficiently achieving its objectives using a soft-max version of goal-based planning. They represent the sequence of actions that lead to a person's future position using a deterministic Markov Decision Process (MDP) over a grid representing the environment. People do not move in a perfectly predictable manner, so the robot has to reason probabilistically about their future locations. By maximizing the entropy of the distribution of trajectories, which are subject to the constraint of matching the reward of the person's behavior in expectation, they obtain a distribution over trajectories. One interesting feature is the fact that the feature-based cost function learned using this approach allows accurate generalization to changes in the environment. Although to successfully predict the future trajectory of a person through an environment the authors require a setting where the human behavior is fully observable and not very crowded. In another approach Ryoo and Aggarwal (2006) tries to construct an intelligent system which will perform early recognition from live video streams in real-time. They introduce two new human activity prediction approaches which are able to cope with videos from unfinished activities. Integral bag-of-words is a probabilistic activity prediction approach that constructs integral histograms to represent human activities. Simply put, the idea is to measure the similarity between a video and the activity model by comparing their histogram representations. The other approach is called Dynamic bag-of-words which considers the sequential nature of human activities, while maintaining the bag-of-words advantages to handle noisy observation. The motivation is to divide the activity model and the observed sequence

into multiple segments to find the structural similarity between them. That is, the bag-of-words paradigm is applied to match the interval segments, while the segments themselves are sequentially organized based on their recursive activity prediction formulation. They've managed to correctly predict ongoing activities even when the videos provided contain less than the first half of the activity.

Kitani et al. (2012) address the task of inferring the future actions of people while modeling the effect of the physical environment on the choice of human actions with prior knowledge of goals. They have focused on the problem of trajectory-based human activity analysis exploring the interplay between features of the environment and pedestrian trajectories. To integrate the aspects of prior knowledge into modeling human activity, they have leveraged recent progress in semantic scene labeling and inverse optimal control. Semantic scene labeling provides a way to recognize physical scene features such as pavement, grass, tree, building and cars, playing a critical role in advancing the representational power of human activity models. The authors propose a Hidden variable Markov Decision Process (HMDP) model which incorporates uncertainty (e.g., probabilistic physical scene features) and noisy observations (e.g., imperfect tracker) into the activity model to express the dynamics of the decision-making process. Since the proposed method encapsulates activities in terms of physical scene features and not physical location, it is also able to generalize to novel scenes transferring knowledge. They are able to forecast possible destinations of the pedestrians through a path, but this evaluation is limited to the physical features of the environments.

Li et al. (2012) propose a framework for long-duration, complex activity, prediction by discovering the causal relationships between constituent actions and the predictable characteristics of activities. This approach uses the observed action units as context to predict the next possible action unit, or predict the intention and effect of the whole activity. The efficiency of their method was tested on the complex activity of playing a tennis game and predicting who will win the game with a relative success (0.65 of certainty with 60% of observed game). Recently Koppula and Saxena (2016) developed a framework where their goal is to enable robots to predict the future activities as well as the details of how a human is going

to perform them in short-term (e.g., 1-10 seconds). With an anticipatory temporal conditional random field (ATCRF), they start modeling the past with a standard CRF but augmented with the trajectories and with nodes/edges representing the object affordances, sub-activities, and trajectories in the future. Their algorithm obtains an activity anticipation accuracy of 84.1%, 74.4% and 62.2% for 1, 3 and 10 seconds of anticipation. We refer the reader to Aggarwal and Ryoo (2011); Vishwakarma and Agrawal (2013); Onofri et al. (2016) for in-depth surveys of the field.

2.3 Datasets

With the release of Microsoft Kinect, several research groups collected different datasets to perform research on 3D action recognition and to evaluate different methods in this field. An initial research was conducted to analyze several datasets from different sources like MSR-Action3D (Li et al., 2010) which was one of the earliest ones which started the research in depth-based action analysis and contained depth samples of sequences of gaming actions e.g. forward punch, side-boxing, forward kick, side kick, tennis swing, tennis serve, golf swing, etc. MSRDailyActivity3D dataset (Wang et al., 2012) contains 320 samples of 16 daily activities with higher intra-class variation: drink, eat, read book, call cellphone, write on a paper, use laptop, use vacuum cleaner, cheer up, sit still, toss paper, play game, lay down on sofa, walk, play guitar, stand up, sit down. Also the LIRIS (Laboratoire d'InfoRmatique en Image et Systèmes d'information) dataset (Wolf et al., 2014) which contains (gray/rgb/depth) videos of people performing various activities taken from daily life (discussing, telephone calls, giving an item etc.). The CMU (Carnegie Mellon University) MoCap dataset[1] contains marker positions and skeleton movement capture using motion capture techniques. It contains actions of a variety of categories like Human Interaction, Interaction with Environment, Locomotion Physical Activities and Sports, Situations and Scenarios and finally Test Motions.

[1] http://mocap.cs.cmu.edu/

In a latter phase of our research we discovered the CAD-60 (Sung et al., 2011) and the CAD-120 (Koppula et al., 2013) dataset [2]. The CAD-60 features: 60 RGB-D videos; 4 subjects: two male, two female, one left-handed; 5 different environments: office, kitchen, bedroom, bathroom, and living room; 12 activities: rinsing mouth, brushing teeth, wearing contact lens, talking on the phone, drinking water, opening pill container, cooking (chopping), cooking (stirring), talking on couch, relaxing on couch, writing on white-board, working on computer. The CAD-120 features: 120 RGB-D videos of long daily activities; 4 subjects: two male, two female, one left-handed; 10 high-level activities: making cereal, taking medicine, stacking objects, unstacking objects, microwaving food, picking objects, cleaning objects, taking food, arranging objects, having a meal; 10 sub-activity labels: reaching, moving, pouring, eating, drinking, opening, placing, closing, scrubbing, null; 12 object affordance labels: reachable, movable, pourable, pour to, containable, drinkable, openable, placeable, closable, scrubbable, scrubber, stationary and also containing tracked skeletons. Like our dataset, CAD-120 is the only dataset that we found which contains long sequences of actions, thus we will compare some our methods to see if they scale to other domain of actions.

The most recent recorded dataset that we are aware of is NTU RGB+D (Shahroudy et al., 2016) which is a large-scale dataset for RGB+D human action recognition with more than 56 thousands of video samples and 4 million frames, collected from 40 distinct subjects. This dataset is the only dataset recorded with Kinect v2 and contains 60 different action classes including daily, mutual, and health-related actions. Unfortunately, this dataset was discovered in a very late phase of our research so it was impossible for us to use it in our experiments. Table 2.1 shows the comparison between some of the discussed datasets amongst others with our PRECOG RGB+D dataset.

[2] http://pr.cs.cornell.edu/humanactivities/data.php

TABLE 2.1: Comparison between publicly available RGB-D human activity datasets. The majority of the datasets were recorded with the Kinect V1 sensor.

Datasets	Samples	Classes	Subjects	Views	Sensor	Modalities	Year
MSR-Action3D	567	20	10	1	N/A	D+3DJoints	2010
CAD-60	60	12	4	-	Kinect v1	RGB+D+3DJoints	2011
RGBD-HuDaAct	1189	13	30	1	Kinect v1	RGB+D	2011
MSRDailyActivity3D	320	16	10	1	Kinect v1	RGB+D+3DJoints	2012
Act42	6844	14	24	4	Kinect v1	RGB+D	2012
CAD-120	120	10+10	4	-	Kinect v1	RGB+D+3DJoints	2013
3D Action Pairs	360	12	10	1	Kinect v1	RGB+D+3DJoints	2013
Multiview 3D Event	3815	8	8	3	Kinect v1	RGB+D+3DJoints	2013
PRECOG	**360**	**8**	**12**	**1**	**Kinect v1**	**RGB+D+3DJoints**	**2014**
Online RGB+D Action	336	7	24	1	Kinect v1	RGB+D+3DJoints	2014
Northwestern-UCLA	1475	10	10	3	Kinect v1	RGB+D+3DJoints	2014
UWA3D Multiview	900	30	10	1	Kinect v1	RGB+D+3DJoints	2014
Office Activity	1180	20	10	3	Kinect v1	RGB+D	2014
UTD-MHAD	861	27	8	1	Kinect v1	RGB+D+3DJoints+ID	2015
UWA3D Multiview II	1075	30	10	5	Kinect v1	RGB+D+3DJoints	2015
NTU RGB+D	56880	60	40	80	Kinect v2	RGB+D+IR+3DJoints	2016

2.4 RGB-D Sensors

RGB-D sensors lead to a boost of new applications in the field of 3D vision. These sensors allied with machine learning techniques to aid in feature representation and decision making can be used for 3D mapping and localization, navigation, path planning, object recognition and human tracking. An example of this is the skeleton tracking algorithm associated with Kinect, which employs a random decision forest trained with 1 million examples to infer the position of the joints of the human body. This is one of many possibilities that apply intelligent machine learning techniques to this new type of data obtained by these sensors. In this chapter we discuss the existing RGB-D sensors, some of the publicly available datasets and the PRECOG dataset recorded by us also available to the public.

RGB-D sensors combine RGB color information with per-pixel depth information providing a much more similar input to our own senses than a simple 2D

image. Nowadays consumer RGB-D sensors cost less than $200. Although these sensors have existed for years (Swiss Ranger SR4000 [3], PMD Tech products [4]), the cost of these sensors was around $10,000 each, making it impossible for its mass dissemination.

Several low-cost RGB-D sensors are now available to the consumer such as, Asus Xtion PRO LIVE, Intel RealSense Series, Structure Sensor and Microsoft Kinect V1 and V2. Kinect V1 (Figure 2.3) was released in November 2010 (with per-pixel depth sensing technology developed by PrimeSense), creating a significant change in the use of gaming devices in the end-consumer market. After a preview at the E3 game convention in the Windows Media Centre Environment, the product was made available in North America on November 4, 2010 and more than 24 million units have been sold.

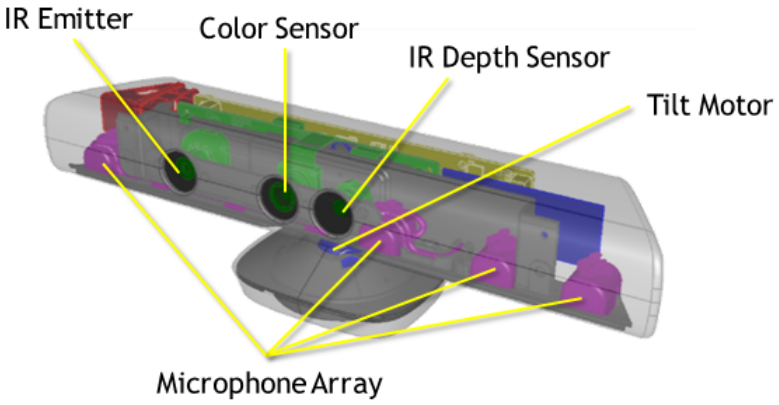

FIGURE 2.3: Kinect for Windows Sensor Components which allows the sensor to capture RGB and depth frames

Source: Microsoft Corporation (2016b)

Kinect performs skeletal tracking which allows the sensor to recognize people and follow their actions. Using the infrared (IR) camera, Kinect can recognize up to six users in the field of view of the sensor. Of these, up to two users can be tracked in detail. An application can locate the joints (Figure 2.4) of the tracked users in space and track their movements over time.

[3] http://hptg.com/industrial/
[4] http://www.pmdtec.com/

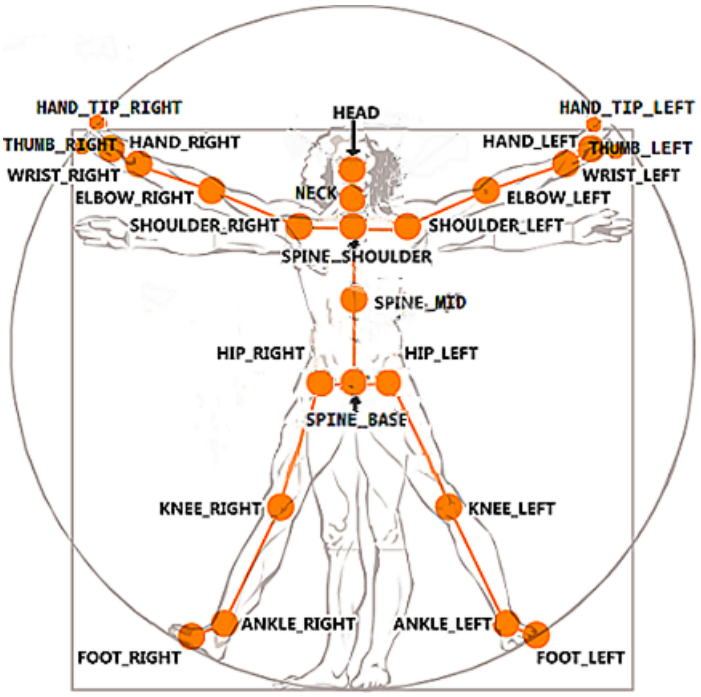

FIGURE 2.4: Skeleton joints of the human body that are captured by the Kinect sensor and can be accessed through the Kinect SDK

Source: Microsoft Corporation (2016c)

For each frame, the depth image captured is processed by the Kinect runtime into skeleton data. Skeleton data contains 3D position data. The position of a skeleton and each of the skeleton joints (if active tracking is enabled) are stored as (x, y, z) coordinates. Skeleton space coordinates are expressed in meters. The x-, y-, and z-axes are the body axes of the depth sensor as shown in Figure 2.5. This feature will be the most relevant for our goal of human activity recognition and prediction.

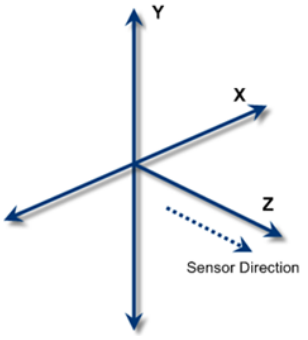

FIGURE 2.5: Kinect right-handed coordinate system used to describe the positions of the skeleton joints in 3D

Source: Microsoft Corporation (2016a)

In 2014 Microsoft released the second-generation Kinect for Windows (Figure 2.6), based on the same core technology as Kinect for Xbox One. It has greater accuracy with three times the fidelity of its predecessor and can track without visible light by using an active IR sensor. It has a 60% wider field of vision that can detect a user up to 3 feet from the sensor, compared to six feet for the original Kinect, and can track up to 6 skeletons at once.

FIGURE 2.6: Kinect V2 sensor

Source: Microsoft Corporation (2017)

2.5 Discussion

The complexity and ambiguity of human activity combined with uncontrolled environments, explains the difficulty of deploying human activity recognition systems into real-word scenarios. Recently, some of that complexity has been reduced due to the recent rise of low-cost RGB-D sensors which allowed researchers to abstract from computer vision techniques and focus on modeling human behavior alone. Nevertheless there is still room for improvement.

Manually labeled datasets are required to perform supervised learning. In order to simplify the cumbersome task of labeling human activity datasets we propose a clustering-based method for semi-unsupervised labeling of human activity. We also demonstrate that it is possible to implement a system capable of recognizing

human actions in sequences of actions with results that overcome the current state of the art (see Chapter 4) and perform action prediction (see Chapter 5) of the next immediate action by discovering patterns in hierarchical structures (high-level sequences containing low-level atomic actions) unlike existing state of the approaches that perform early recognition (action recognition before the action is completed) or prediction ahead in time in seconds.

Chapter 3

Methodology

"The roots of education are bitter, but the fruit is sweet."

Aristotle

In this chapter, we describe our conceptual approach for the development of an activity recognition and prediction system. The proposed approach addresses key issues in the field of HAR, namely the dependence on human judges to label large amounts of data required to train the classifiers and create the models, the performance of existing activity recognition systems which can still be improved and finally, the creation of a solution that combines action recognition with action prediction in sequences of actions.

This chapter is organized in six distinct sections that describe the following topics: (i) an overview of the solution concept and the definitions, (ii) how to reduce the human input required for labeling data, (iii) how to perform real-time action recognition and how to make it as early as possible, (iv) perform structured prediction in sequences/patterns of actions, (v) the release of the public PRECOG dataset and (vi) the potential advantages and shortcomings of the proposed approach.

3.1 Solution Concept and Definitions

Computer science has a wide array of methodological approaches (Peffers et al., 2007). For this specific project we will adopt a module-oriented build-and-test approach. This approach consists in identifying a necessity for a given software module to be implemented, which algorithms should be used, the actual implementation of the module and exhaustive testing with adequate data and validation of the results.

Based on the proposed methodology a conceptual approach for the development of the activity recognition and prediction system has been defined. The solution was designed with a hierarchical model in mind. This model is illustrated in Figure 3.1.

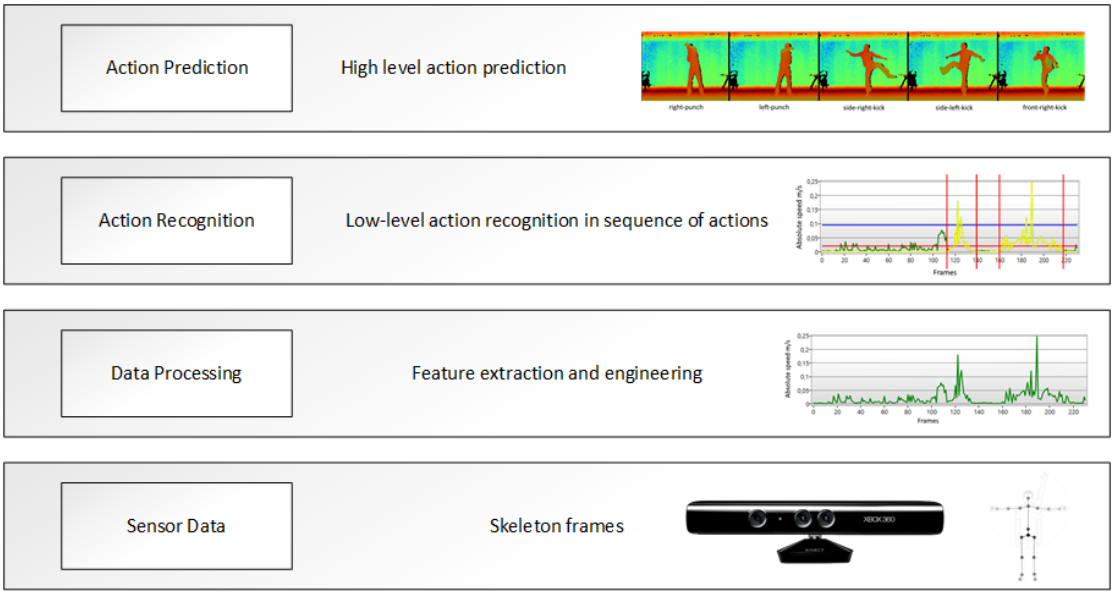

FIGURE 3.1: Conceptual hierarchical model used to describe the action recognition and prediction process with several layers of abstraction.

The action recognition and prediction process is described by the hierarchical model through four levels of abstraction with a *bottom-up* approach, ranging from low-level data acquisition from the sensor, feature extraction and feature engineering to design the input of the classification algorithms, low-level or atomic action recognition to high-level action prediction. This solution has the advantage of facilitating and guiding the development of the system in layers where each layer can

have several exchangeable modules, maximizing the component reutilization. We adopted the following feature extraction method (Figure 3.2). For every skeleton frame sf_i, a feature vector fv_i representing the movement of the human subject is computed. Each skeleton joint J_i has its position defined by a three-dimensional vector [X, Y, Z] that will be used to compute several features. The number of selected joints J_n and the number of computed features f_n will vary depending on the purpose and application.

$$
\begin{array}{lcll}
sf1 = [J_0, J_1, J_2, J_3, ..., J_n] & & sf_i = i\text{th frame} & fv1 = [f_0, f_1, f_2, f_3, ..., f_n] \\
sf2 = [J_0, J_1, J_2, J_3, ..., J_n] & & J_i = i\text{th joint position in 3D} & fv2 = [f_0, f_1, f_2, f_3, ..., f_n] \\
sf3 = [J_0, J_1, J_2, J_3, ..., J_n] & \rightarrow & f_i = \text{feature, i} = 1, 2, ...N & \rightarrow \quad fv3 = [f_0, f_1, f_2, f_3, ..., f_n] \\
\vdots & & fvi = [f_0, f_1, f_2, f_3, ..., f_n] & \vdots \\
sfN = [J_0, J_1, J_2, J_3, ..., J_n] & & & fvN = [f_0, f_1, f_2, f_3, ..., f_n]
\end{array}
$$

FIGURE 3.2: For every skeleton frame sf_i and selected joint J_i we compute a feature f_i that will be added to the corresponding feature vector fv_i.

Below, we describe the key terminology and concepts used in our approach:

Action: The literal definition of an action is the fact or process of doing something, typically to achieve a goal. In this particular case the word action is used to describe a single human motion performed by one or more body parts. For example picking up a glass can be considered an action.

Sequence of actions: A sequence of actions describes a high-level human behavior that it is composed of multiple (low-level) actions. This aggregation of low-level actions compose repeatable patterns that can be identified and used to perform prediction. For example preparing breakfast, is composed of several actions, such as picking up a bowl, pouring cereals and then pouring milk.

Action recognition: This describes the process of recognizing low-level actions performed by a human, with the help of machine learning methods, that were trained to classify actions from a set of features computed from skeleton frames data.

Early action recognition: Early action recognition as the name implies is the process of recognizing an action as soon as possible and preferably before the execution of the action ends. Some of the initial pursuits in action recognition only recognized the action after its execution (Dollar et al., 2005; Gorelick et al., 2007). A system with this feature is useful in a wide scope of scenarios and applications where response time is critical, for example in a surveillance scenario.

Action prediction: Action prediction is the process of predicting the next action that will be executed by a human subject in any given moment in time. The predicted action should be a part of a sequence of actions which as previously said, compose repeatable patterns that are identifiable by the system.

3.2 Semi-unsupervised Labeling of RGB-D videos

Machine learning can be divided into unsupervised learning and supervised learning. Unsupervised learning does not require labeled data, whereas supervised learning does. Labeled data consists of unlabeled data associated with a label, tag or class which provides useful information. The labeling of data is often done by human judges that are asked to assign a label to a piece of unlabeled data. Despite the increase of human understanding methods, many problems still remain open, including modeling of human poses, handling occlusions, and annotating data. One of the tasks that we had to perform when we recorded our dataset with the Kinect sensor was to manually label the sequences frame-by-frame. Manual analysis of video is labor intensive, fatiguing, and error prone.

Given these issues, we propose a method with the pipeline illustrated in Figure 3.3 that tries to simplify and automate the data labeling process, reducing the amount of input required by a human to label a dataset of human activity and consequently model and train a classifier capable of recognizing human activity in real-time. The proposed method will be divided in the following steps: (i) temporal segmentation of sequences of actions, (ii) clustering of temporal segments

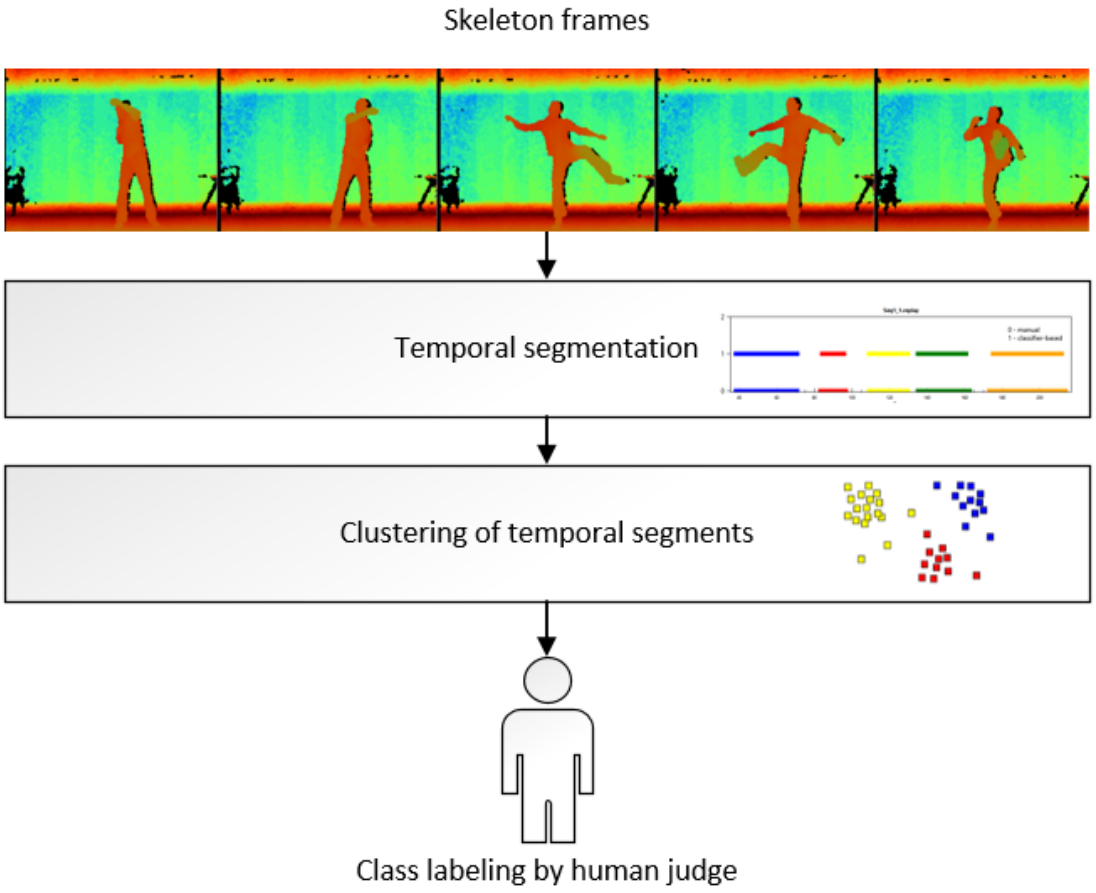

FIGURE 3.3: Proposed pipeline for reducing the amount of input required by a human judge to label a dataset of human activity.

representing an action and (iii) cluster to class association performed by a human judge.

Temporal segmentation: is a process which consists in dividing sequences of actions into well-defined segments which represent an action as illustrated in Figure 3.4 and it is very important to understand and build computational models of human activity (Zhou et al., 2013). Ideally, temporal segmentation will group frames that are part of a specific action into one segment with clear boundaries that will represent a complete action. Temporal segmentation presents several challenges: the variability in the temporal scale of human actions, overlapping of the current action with the next, the complexity of representing articulated motion, and the exponential nature of all possible movement combinations.

The considered datasets for our experiments contain sequences of actions, so our very first task was to devise a method which automatically decomposes the

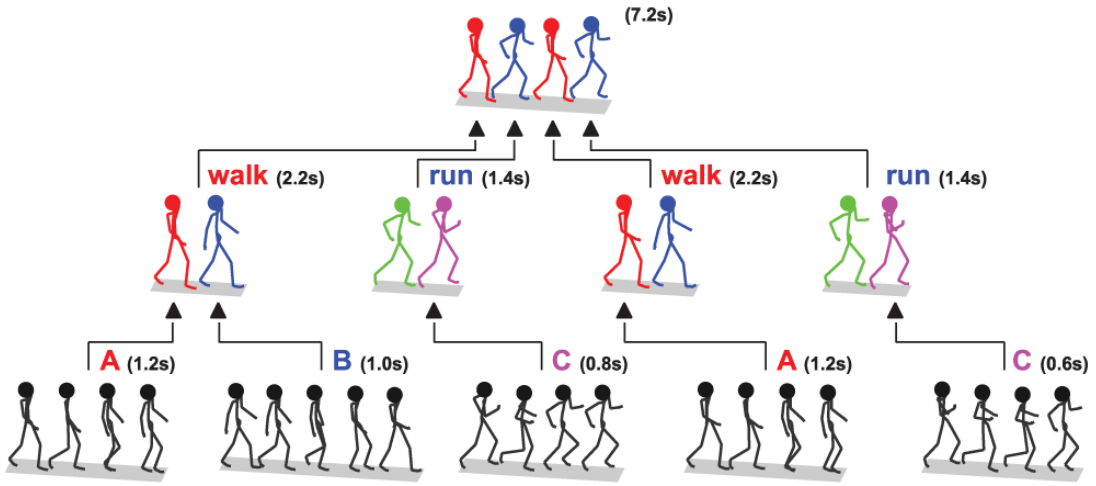

FIGURE 3.4: Example of a motion capture sequence segmentation by Zhou et al. (2013) where a full sequence of actions is decomposed into simpler actions.

sequence in temporal segments where each segment represents an isolated action. We propose three distinct approaches for temporal segmentation. The first approach is unsupervised (application oriented), based on heuristics and uses the absolute velocity of the skeleton joints to create temporal segments. The second is also unsupervised and it is based on Warped K-means (WKM) which is a general purpose segmentation-based partitional clustering procedure. The final approach is a supervised method which uses different kinds of models trained to recognize the neighboring frames between any two actions in a sequence.

Clustering of temporal segments: consists in using a clustering algorithm to group all the temporal segments found by our temporal segmentation method into different clusters which will represent different actions, based on the similarity of the movements performed by the human subject. In order to perform clustering of the temporal segments, first we have to sample the segments for several joints in order to generate the data that will be used for clustering. Figure 3.5 illustrates how we propose to perform sampling.

Given a temporal segment like the one in the yellow shaded region, we take a snapshot of that region of four selected joints (wrist-right; wrist-left; ankle-right; ankle-left), then using our motion measuring method we assign the most active joint of the skeleton to that segment, in this case it is the wrist-right. This procedure is replicated for all the temporal segments found, and ideally it will

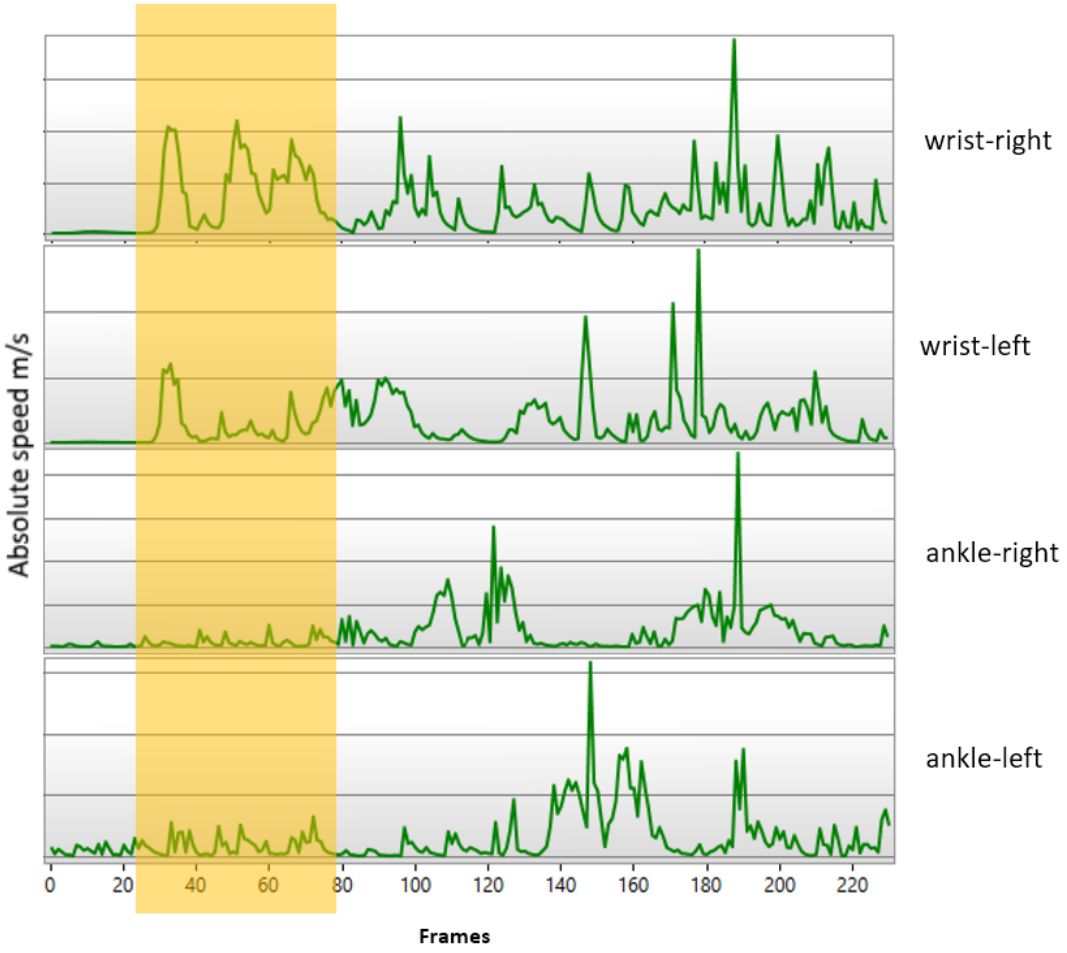

FIGURE 3.5: Example of a sampled temporal segment represented by a vertical cut for all the selected joints.

correspond to the number of actions that compose the sequence. The number of actions found is ignored and will not be used as a feature to allow the system to abstract to scenarios where sequences contain a variable number of actions. The sampling procedure can be portrayed as stacking the joints time-line one on top of another and making vertical slices to extract samples of data that correspond to temporal segments where an action has occurred. For each sampled segment, several features will be computed to create a feature vector that will be fed to several clustering algorithms. In order to facilitate the clustering process we will apply a body filtering method where its purpose is to find the most active joints of the segment and make a decision whether the action is being performed mostly by upper body joints or lower body joints. This will be useful to reduce the confusion between actions where multiple joints are moving at the same time. The expected

result is that in all the samples created from the dataset, patterns will appear that could be clustered by similarity corresponding to the same or similar actions.

Cluster to class association: A human judge will be required to identify the corresponding class of a cluster which represents an action. This classification will be propagated through the entire dataset, replacing the assigned cluster with the identified class by the human judge. Instead of manually labeling each and every frame of the captured data, the human judge only has to label the different classes that were automatically found by our proposed method. A summarized description of the semi-unsupervised labeling process is described by Algorithm 1.

Algorithm 1 High-level description of the semi-unsupervised labeling process of RGB-D videos containing human activity

```
1: procedure SEMI-UNSUPERVISED LABELING OF RGB-D VIDEOS(sequences)
2:     for all <sequences> do
3:         perform temporal segmentation
4:         sample sequence based on temporal segmentation
5:         group samples based on the most active joint
6:         feature vector ← compute features for each sample
7:         training data ← add feature vector to training data
8:     end for
9:     for all <most active joints> do
10:        apply body filtering
11:        clusters ← execute clustering algorithm (training data)
12:        apply cluster labels to the temporal segments
13:    end for
14:    manually assign an action class to a cluster of sampled temporal segments
15: end procedure
```

3.3 Real-time Action Recognition

According to Aggarwal and Ryoo (2011), human activity can be categorized into four different levels: gestures, actions, interactions and group activities. We are interested in recognizing high-level human activities. Most of the past research in human activity recognition has focused on recognizing human activity from still images or from 2D videos (Gavrila, 1999; Oliver et al., 2002; Niu et al., 2004). Estimating the human pose over shorter time scales is the primary focus of these works. Currently, having access to a 3D camera which provides RGB-D videos enables us to robustly estimate human poses and use this information for learning complex human activities. We propose two distinct methods to perform action recognition: (i) temporal segment action recognition and (ii) frame-by-frame action recognition. Both will use the position of the skeleton joints extracted from the skeleton frames which are captured by the Kinect sensor and then, with the help of supervised learning methods we will train a machine learning algorithm to classify different human actions. The proposed method to perform action recognition should be able of handling variations in distance between the body and the Kinect sensor, skeleton orientation, and speed of the actions.

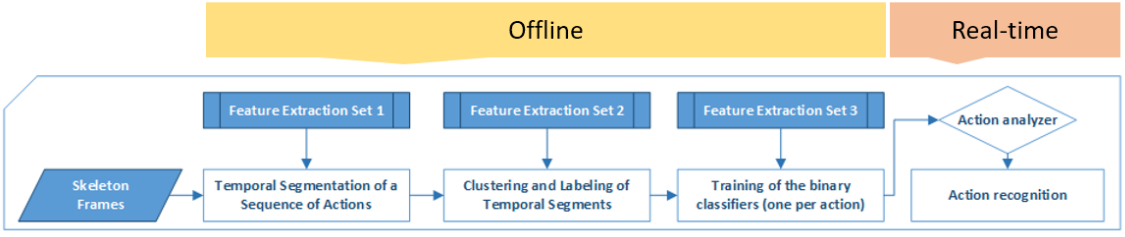

FIGURE 3.6: Pipeline of the modular framework designed to perform action recognition. Offline modules are used to process the data and extract the features that will be used to train the classifiers with different feature sets for each module. The online modules are used to perform real-time action recognition.

Figure 3.6 illustrates how we designed our framework considering the already addressed tasks of temporal segmentation, clustering and labeling, with the addition of new modules for training of the classifiers and for action recognition. Choosing a modular framework allowed us to isolate each task in one module giving us the ability to fully replace a module if required or if a better solution is

presented. Although the Kinect sensor provides raw data containing the color and depth values for every pixel in the video, we will not use that information. We will rely solely on the Kinect skeletal tracking for obtaining the skeleton frames which contain the locations of the various joints of the human skeleton being tracked in 3D. From the locations of the skeleton joints we will compute several metrics that will be used as features, like the absolute speed of each joint, velocity vector of each joint, accumulated displacement of each joint, flexion and extension angle between two bones and bone orientation. Since every module is oriented to a specific problem, the set of extracted features has to be specific. For example, to cluster temporal segments average values which represent the whole temporal segment will be computed, where for real-time action recognition, the features will be computed for every frame. The first three modules of the framework work off-line and are responsible for processing and preparing the data that will be used to train several classifiers. The last module runs in real-time and is the module responsible for recognizing a given action.

Temporal segment action recognition: This approach consists in classifying the actions by their temporal segment as a whole, similar to the research from Cippitelli et al. (2016) where a feature vector representing the whole activity is created and used for classification. Each instance of the feature vector is created by averaging the values of the features for all the frames within each temporal segment. The main stages of this approach which are described in detail in Chapter 4, are:

- Movement detection. If the average movement of all the joints of the skeleton are above a threshold value, then the subject is moving. The threshold value was calculated by averaging the minimum movement value for all the sequences.
- Skeleton Features Extraction. The coordinates of the skeleton joints and bones orientations which represent human postures are extracted to compute several activity features.

- Temporal Segment Creation. Several skeleton frames are stored to create a temporal segment with a length that corresponds to the average length computed from all the temporal segments in the dataset.
- Activity Features Computation. A feature vector representing the average values of all the features from the skeleton frames which are comprised in the temporal segment.
- Classification. The classification is performed using binary classifiers, where each classifier was trained to recognize a specific action which occurs over the duration of a temporal segment.

A possible downside of this approach, is that requires a minimum amount of skeleton frames in order to create the temporal segment that will be used to generate the feature vector for classification. This could mean that the response time in which the system performs the classification will suffer a delay of several frames. Algorithm 2 describes in pseudocode how the action classification for temporal segments is performed.

Algorithm 2 High-level description of the process to perform action recognition of temporal segments.

```
1: procedure TEMPORAL SEGMENT ACTION RECOGNITION(skeletonframe)
2:     if <subject is moving> then
3:         frames ← add skeleton frame
4:         if <has enough frames> then
5:             temporal segment ← create temporal segment from frames
6:             action feature vector ← compute feature vector from the temporal segment
7:             classification ← classify activity feature vector
8:             return classification
9:         end if
10:    end if
11: end procedure
```

Frame-by-frame action recognition: Is our proposal for real-time activity recognition which starts from skeleton joints and computes a vector of features for *each* captured frame from the Kinect sensor. Again we use binary classifiers for recognition purposes, where each class represents a different action. The features and the feature vector computed in this method are the same as in the previous method. The training data generated for each classifier contains frame instances of a given action labeled as positive examples and also frame instances for the remaining actions labeled as negative examples. The main stages of this method are:

- Movement detection (same as before).
- Skeleton features extraction (same as before).
- Frame-by-frame features computation. A feature vector representing the action at a given skeleton frame is created and used for classification.
- Classification. The classification is performed using multi-class and binary classifiers which implement a voting selection approach to select the best possible classification.

The first step of the action recognition process is to measure the average displacement of all the joints of the skeleton to decide if the subject is moving. If the subject is moving an instance of the feature vector is computed from the captured skeleton frame from the Kinect sensor and fed to the classifiers. From all the correct classifications we select the classification with the lowest error. This is described in more detail in Algorithm 3.

3.4 Real-time Action Prediction

Although recent years have seen an increase of research and development in systems specialized in human activity recognition (Wolf et al., 2014; Koppula et al.,

Algorithm 3 High-level description of our method to perform frame-by-frame action recognition implementing a best action voting strategy.

```
1: procedure FRAME-BY-FRAME ACTION RECOGNITION(skeletonframe)
2:     if <subject is moving> then
3:         action feature vector ← compute features from the skeleton frame
4:         for all binary classifiers do
5:             classification ← classify activity feature vector
6:             store classification
7:         end for
8:         best classification ← select correct classification with lowest error rate
9:         return best classification
10:    else
11:        return subject not moving
12:    end if
13: end procedure
```

2013; Ramirez-Amaro et al., 2015; Wachter and Asfour, 2015; Gaglio et al., 2015; Cippitelli et al., 2016; Onofri et al., 2016) the same cannot be said about systems specialized in human activity prediction (Ryoo, 2011; Kitani et al., 2012; Koppula and Saxena, 2016). This is the point in which our research distinguishes itself from most of the existing approaches. We propose a system capable of not only recognizing but also predicting human activity using machine learning techniques. Our goal is to recognize as early as possible the current action being performed by a human and predict the next/future activity that will occur. For example, in a industrial manufacturing scenario, if a robot can anticipate the actions of a human worker, it can provide him tools and parts when he requires them.

We would like to perform action prediction in the context of long activities or sequences of actions which have a hierarchical structure where a sequence is composed by several actions. By knowing the sub-activities performed in the past and the hierarchical structure of the activities we expect that we can predict the next action that will occur in the sequence. We propose two methods to perform action prediction: (i) N-gram action prediction where we will perform experiments with several machine learning classifiers like: Multilayer Perceptron (MLP) (Kubat, 1999), SVM using pairwise classification (Platt, 1998), RF with n-grams of variable size (Breiman, 2001) and (ii) CRF based approach, which are mostly used in Natural Language Processing (NLP) problems. CRF are suited

for labeling structured data, they model rich contextual relations and are capable of learning and inferring a small and discrete label space such as our sequences of actions (Blei et al., 2004). CRF have been used in a parallel work to model the sequential nature of actions in a sequence (Koppula and Saxena, 2016), but where other approaches try to predict an outcome or anticipate ahead in time (seconds), we try to predict what will be the next action of a subject. There are several activities where a human subject performs certain actions as a sequence of actions. With that premise we would like to prove that given the observations of a scene containing a human performing an action a for time t, it is possible to predict the possible action $a+1$ in a sequence of actions, obtaining a distribution over the future possibilities (Figure 3.7). These observations will be captured by a Kinect sensor which will help us obtain the human pose using Microsoft's skeleton tracker.

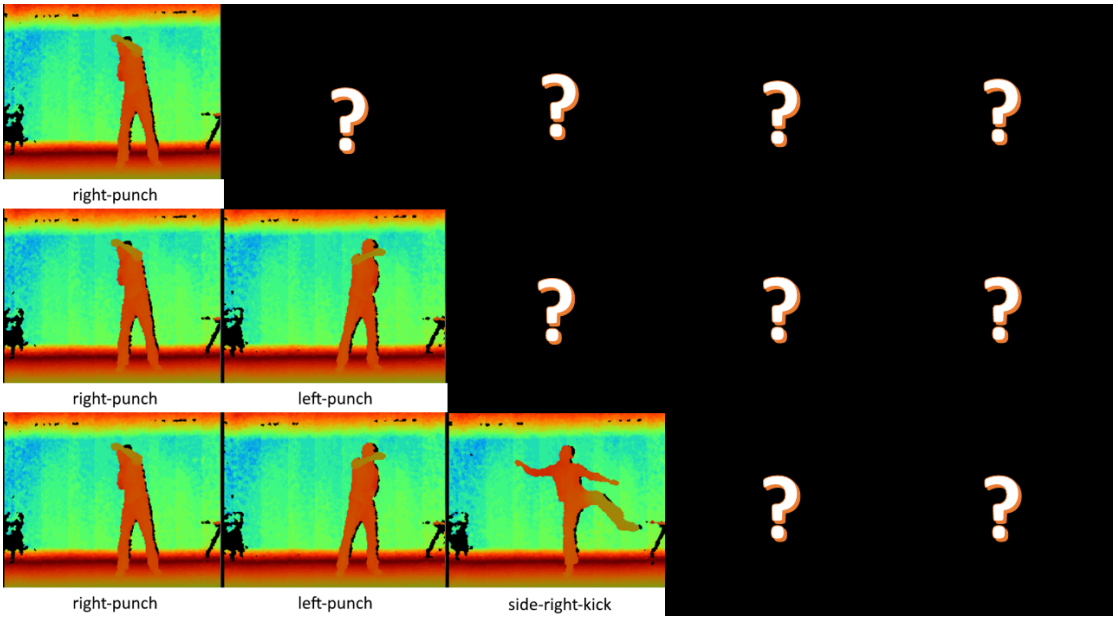

FIGURE 3.7: Illustration of the expected prediction process that the system will perform given the current recognized action and the history of recognized actions.

The goal is to label the history of previous actions with a tag that represents the next possible action. Manually labeled data and data labeled by our binary action recognition classifiers will be used to create our training set that will be fed to the different prediction classifiers.

Figure 3.8 illustrates the pipeline of the modular framework capable of performing action recognition and prediction. An (*action prediction*) module will be implemented, responsible for the prediction of the next action that will occur based on the information received from the action recognition module of the current action being observed. The action prediction module, will work in real-time, keeping track of the history of the actions being recognized and will generate the feature vector for the prediction classifier based on that history.

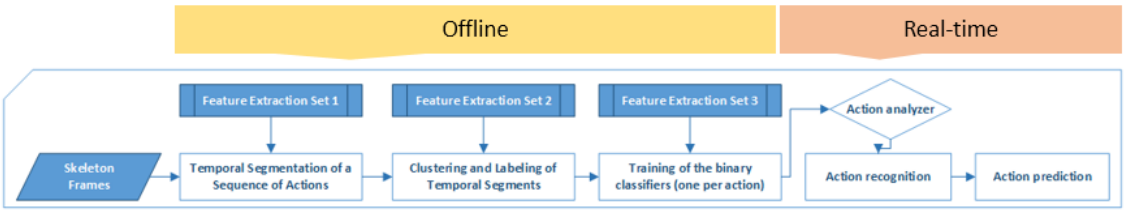

FIGURE 3.8: Complete pipeline of the modular framework designed to perform action prediction. The difference here is the addition of a new on-line module responsible for performing real-time action prediction.

Given the history of actions the prediction classifier has to predict the label of the future action that will occur in that sequence. Algorithm 4 gives an overview of the prediction process.

Algorithm 4 High-level description of the action prediction process which is capable of predicting actions based on the current recognized action and the history of actions recognized.

```
1: procedure ACTION PREDICTION(skeletonframe)
2:     if <subject is moving> then
3:         activity feature vector ← compute features from the skeleton frame
4:         for all binary classifiers do
5:             action classification ← classify activity feature vector
6:             store action classification
7:         end for
8:         best classification ← select correct classification with lowest error rate
9:         actions history ← add best classification action
10:        prediction feature vector ← compute feature vector from history of actions
11:        prediction classification ← classify prediction feature vector
12:        return prediction
13:    else
14:        return subject not moving
15:    end if
16: end procedure
```

N-Gram Action Prediction: An n-gram model is a type of probabilistic language model for predicting the next item in such a sequence in the form of a $(n-1)-order$ Markov model (Jurafsky and Martin, 2009). Widely used in the fields of computational linguistics and probability, an n-gram is an *n-character* slice of a longer sequence of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus. In our case the items will represent a sequence of sub-activities. The n-grams are composed by combinations of the actions of the sequence where the last action is the attribute to be used as the class. For example, the sequence "*right-punch, left-punch, side-right-kick, side-left-kick, front-left-kick*" would compose the following n-grams:

- 3 combinations of the tri-gram: "*right-punch, left-punch, side-right-kick*".

- 4 combinations of the quad-gram: "*right-punch, left-punch, side-right-kick, side-left-kick*".
- 5 combinations of the penta-gram: "*right-punch, left-punch, side-right-kick, side-left-kick, front-left-kick*".

Conditional Random Fields Action Prediction: Conditional random fields (Sutton and McCallum, 2010) model rich contextual relations conditioned on several features as input. Usually CRF are used in computational linguistics (NLP) to perform tasks like: word breaker, POS tagging, and named entity recognition where the goal is to label a sentence (a sequence of words or tokens) with tags like *adjective, noun, preposition, verb, adverb, article*. We propose a parallel approach where a sequence of actions can be seen as a sequence of text and each action is seen as a word. Our intent is to use CRF to predict the next action given the current action performed and the history of actions performed. Just like any classifier, we'll first need to decide on a set of feature functions f_i. In a CRF for our model, each feature function will be a function that takes as input:

- a sequence of actions s.
- the position i of a action in the sequence.
- the label l_i of the current word.
- the label h_i of the history of the previous actions.

By restricting our features to depend on only the current action and on the history of the previous actions, we are building the special case of a linear-chain CRF (Figure 3.9).

To create the training data, we will gather all the existing sequences of actions and for each sequence, we perform all the possible combinations of *current action – history of actions – next action*, computing a distribution over the possible future actions (Table 3.1).

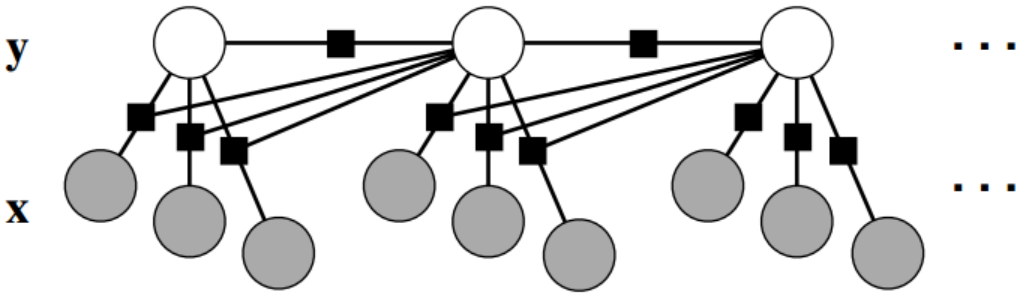

FIGURE 3.9: Graphical model of a linear-chain CRF that models our prediction approach which depends on the current action and the previous recognized actions.

Source: Sutton and McCallum (2010)

TABLE 3.1: Example of the instances generated for the training corpus from a single sequence of actions. This will be repeated for all the sequences of the dataset.

Current action	History of actions	Predicted action
RightPunch	RightPunch	Side LeftKick
SideLeftKick	RightPunch SideLeftKick	LeftPunch
LeftPunch	RightPunch SideLeftKickLeft Punch	SideRightKick
SideRightKick	RightPunch SideLeftKick LeftPunch SideRightKick	FrontRightKick

Each record of the training corpus represents a sequence of actions (like a matrix) and each row describes an action to be predicted. To a column size N, the first $N-1$ columns are used as input data to generate the binary features and train the model. The nth column is the action that the model should predict.

3.5 PRECOG Dataset

When we started this research the majority of the available datasets contained only isolated actions. This represented a problem since we proposed ourselves to perform action recognition and prediction in sequences of actions composed by single actions. We saw this as a chance to create a new dataset that contains

sequences of actions. In a real world situation we expect to have a subject perform a sequence of actions instead of a single isolated action like portrayed in most of the previous discussed datasets. With this in mind we recorded a dataset containing sequences of aggressive actions performed by 12 different subjects, 9 male and 3 female with no fighting experience. We chose the Kinect sensor to record the dataset because since its release it has become the standard to record and collect human activity datasets as shown in Table 2.1. We used the Kinect sensor indoors, fixed in a front-view, with artificial lighting to record the dataset with sequences of combat movements composed of 8 different actions: *right-punch; left-punch; elbow-strike; back-fist; right-front-kick; left-front-kick; right-side-kick; left-side-kick*. Using combinations of those 8 actions we created 6 distinct sequences (each sequence contains 5 actions, Table 3.2). Of the 12 subjects recorded, each subject performed 6 different sequences. A total of 72 sequences, 360 actions were recorded.

TABLE 3.2: Description of the sequences of actions that were captured. The layout of the actions in the sequences was carefully selected in order to guarantee some logical patterns and repetitions in the sequences.

	Action 1	Action 2	Action 3	Action 4	Action 5
1	right punch	left punch	side right kick	side left kick	front right kick
2	right punch	side left kick	left punch	side right kick	front right kick
3	right punch	front left kick	side left kick	back fist	front right kick
4	back fist	left punch	side right kick	side left kick	front left kick
5	back fist	side right kick	right punch	front left kick	side left kick
6	back fist	side right kick	front right kick	elbow strike	side left kick

The data was collected in .xed files which contains RBG, depth and skeleton information (Figure 3.10), we also exported the skeleton data to .csv format where each column of the Comma Separated Value (CSV) file refers to a specific skeleton joint and each row contains the 3D position per joint, for every captured frame. RBG videos were recorded in the resolution of 640 X 480 and the depth frames with a resolution of 320 x 240 (Figure 3.11). Kinect is able to track 20 joints of

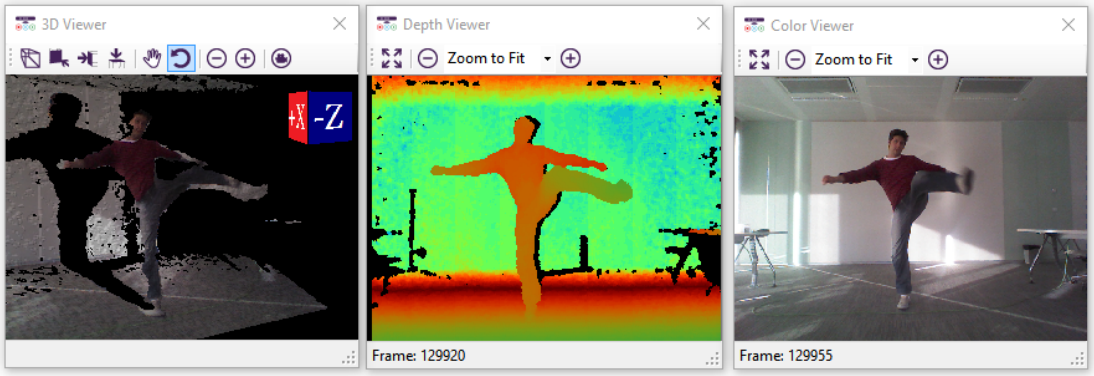

FIGURE 3.10: Microsoft provides visualization tool which allows users to explore the 3D, Depth and RBG view of a recorded .xed file.

a subject's skeleton. Skeleton frames are generated at the rate of 30 frames per second, and each frame consists of the 3D coordinates of 20 body joints along with their tracking states (tracked, inferred, or not tracked). Although we recorded RGB and depth information, our framework relies exclusively on the position of the skeleton joints to extract relevant features.

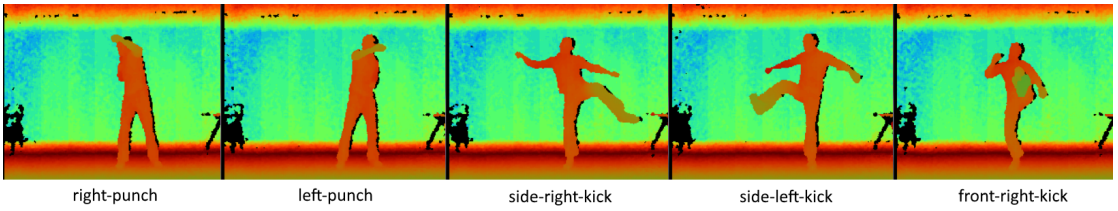

FIGURE 3.11: Selection of five depth frames from a recorded sequence where each frame correspond to a different action from a total of five actions.

The dataset[1] is fully annotated and available for public usage. Along with the dataset we released source code to an application which allows researchers and developers to visualize each sequence of the dataset in real-time (Figure 3.12) while providing a starting tool for development.

3.6 Discussion

Supervised machine learning techniques require representative labeled training sets. In order to obtain those representative training sets, data acquisition is

[1]https://github.com/DavidJardim/precog_dataset_16

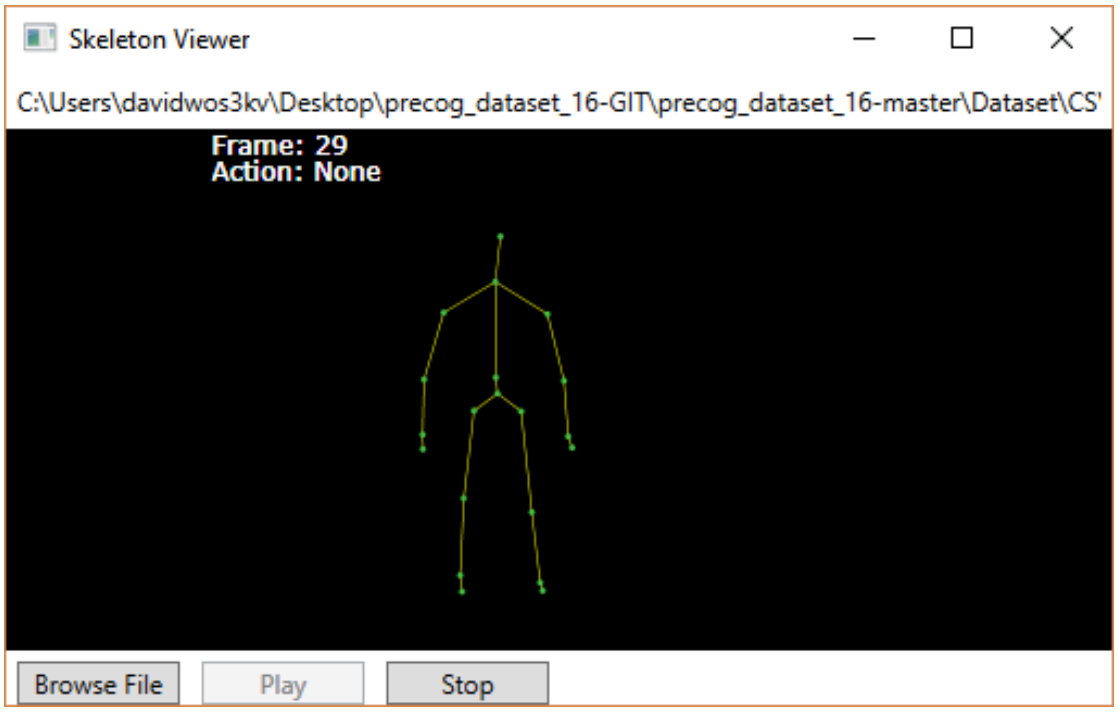

FIGURE 3.12: Visualization tool released with the dataset

required. Data acquisition can be seen as the most expensive task in machine learning, because it requires the collection of the data and then the correct labeling of all the samples which is usually done by human judges. With our proposal for semi-unsupervised labeling of human activity RGB-D videos we expect to reduce the amount of input required by a human judge to label a dataset. We foresee some difficulties in labeling similar or ambiguous actions since we rely only on the skeleton joints and ignore any contextual information. For instance, from the system's point of view what is the difference between picking up a glass or a cellphone?

Our approaches to activity recognition will run in real-time providing a very short response time and will rely only on the information of the skeleton joints. Since they are similar to other approaches in the state of the art, they might find it difficult as well to recognize similar actions. It is unknown how the system will respond if presented with a new activity for which it has never been trained, will it be able to recover and continue recognition? Since it is impossible to train a system for each and every possible and existing action, the detection of unknown activities still remains an issue for this kind of systems.

We are confident that with our prediction approach we will be able to predict the next action that will occur in a sequence of actions as long as the scope of the sequences is limited. It is impossible to foresee all the possible combinations of actions in the real world. This means that this approach has a limited application to scenarios where certain patterns and combinations of actions are expected. This approach might reveal a low intolerance to error since it depends highly on the action recognition module to provide the correct recognized action to generate the history of previous actions and perform the correct prediction.

The PRECOG dataset remains as a very important contribution of this research, since as far as we know, it is one of the few RGB-D datasets that contains long sequences of actions composed by shorter actions. One of its limitations is that it was recorded with a single sensor providing only a front view of the subjects. In a perfect scenario, the recordings should be done with several sensors providing a 360° view of the subject. We could not use the Kinect V2 sensor to perform the data collections because the sensor was not available to the public at the time.

Chapter 4

Semi-unsupervised Labeling and Recognition of Human Activity

"The good life is one inspired by love and guided by knowledge."

Bertrand Russel

In this chapter, we apply our temporal segmentation and clustering methods for semi-unsupervised labeling of RGB-D videos of human activity by modeling the skeleton movement. We also address the problem of real-time human activity recognition using semi-unsupervised and manually labeled training data. This chapter is structured as follows: in Section 4.1 we describe the experimental setup; in Section 4.2 we demonstrate our several temporal segmentation approaches; in Section 4.3 we show how our approach can be used to perform semi-unsupervised labeling of human activity; in Section 4.4 we perform real-time human activity recognition using two different methods (temporal segmentation recognition and frame-by-frame action recognition); and in Section 4.5 we discuss the results.

4.1 Experimental Setup

The performance of our solution for semi-unsupervised labeling and recognition of human activity will be evaluated on two public 3D human activity datasets: PRECOG dataset and the Cornell Activity Dataset 120. The PRECOG dataset features 72 RGB-D video sequences of aggressive actions; 12 subjects; 6 distinct sequences of actions with 8 aggressive actions. The CAD-120 dataset features: 120 RGB-D videos of long daily activities; 4 subjects: two male, two female, one left-handed; 10 high-level activities; 10 sub-activities and 12 object affordance labels. The clustering algorithms will use euclidean distance as the distance function with a maximum of 500 iterations (this value was obtained by verifying experimentally the number of typical iterations for the stabilization of the clusters). The classification methods will be tested by using k-fold cross-validation.

We are using the Kinect for Windows SDK which provides the tools and APIs, both native and managed, required to develop Kinect-enabled applications in C# for Microsoft Windows. We used WEKA (Eibe Frank and Witten, 2016) for the implementations of learning algorithms that we applied to our data. WEKA contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. WEKA can be used as a stand-alone application (Figure 4.1) or as an API imported into a Java project. We had to use IKVM.NET [1] which is an implementation of Java for Mono and the Microsoft .NET Framework. IKVM.NET enables Java and .NET interoperability allowing us to use the WEKA API (Java) in our C# .NET application.

4.2 Temporal Segmentation

In this section we describe the features used to perform temporal segmentation, and how to compute them. Then we present the experimental results for the Absolute speed-based segmentation method, Warped K-means segmentation method

[1] http://www.ikvm.net/

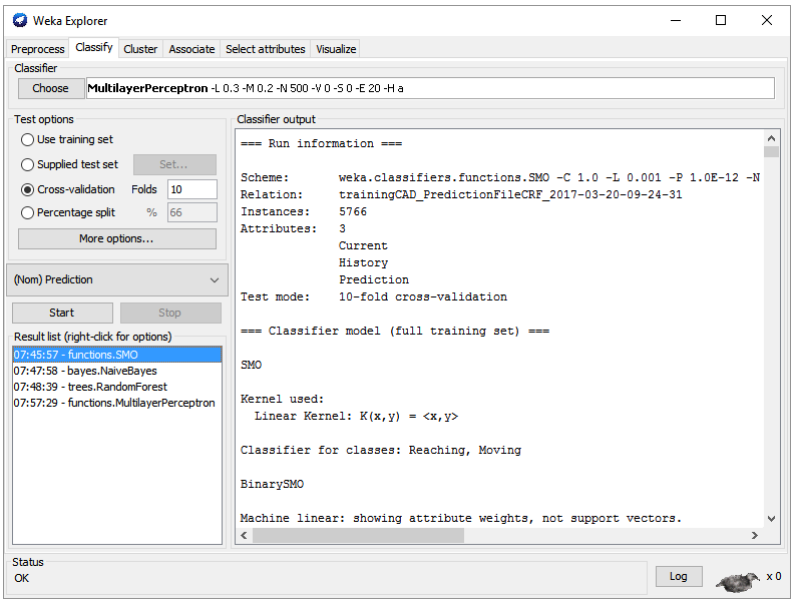

FIGURE 4.1: WEKA Workbench

and the Classifier-based segmentation method, detailed in sub-sections 4.2.2, 4.2.3, 4.2.4 respectively.

4.2.1 Features

Kinect tracks 20 joints from the human body in 3D with X,Y,Z information, of those 20 joints we highlighted five: *spine-base, wrist-right, wrist-left, ankle-right and ankle-left* (Figure 4.2). These five joints were selected to extract features that will be used to perform temporal segmentation. This choice was supported by several approaches from related work (Koppula et al., 2013; Nirjon et al., 2014; Gaglio et al., 2015; Cippitelli et al., 2016) which use similar subsets of joints for evaluation of their proposed algorithms and concluded that it is not required to use all the joints of the skeleton to obtain knowledge from the skeleton movement. The size of the feature vector will affect the accuracy of the classifiers but also the response time. The greater the size of the feature vector, the longer the response time.

The joint positions returned by the Skeletal Tracking (ST) are affected by noise and in some cases joint occlusion might occur. In these situations, when the ST is unable to track a certain joint the joint data is inferred by calculating

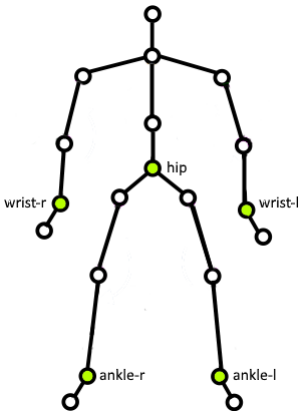

FIGURE 4.2: Illustration of the five skeleton joints selected (in green) to extract features that will be used in the temporal segmentation methods.

it from other tracked joints. Since the data is calculated, confidence in the data is very low. The first step was to obtain the skeleton joints independent of their position in the camera reference frame. To achieve this we considered the spine-base as the origin of the skeleton and computed the position of each joint as the following: $normalizedJointPosition = jointPosition - spineBasePosition$. Based on the normalized joint position, absolute speed is calculated for each joint frame-by-frame. Speed is the distance traveled in meters divided by the time in milliseconds it took to travel this distance (Eq. 4.1).

$$|s_i| = \left| \frac{d(J_i)}{\Delta t} \right|, \; J_i = ith \; joint \; position \; in \; 3D \tag{4.1}$$

Distance (Eq. 4.2) is calculated by the difference of the position in two consecutive frames: $distance = jointPosition(\text{frame n} + 1) - jointPosition(\text{frame n})$. The Kinect sensor captures 30 frames per second (FPS), so $\Delta t = 33,3$ milliseconds.

$$d_i = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{4.2}$$

Figure 4.3 illustrates the absolute speed of the right ankle over time through a sequence of actions. Although not required for our use cases, the skeleton height could also be normalized to handle varying distances of the subjects to the sensor.

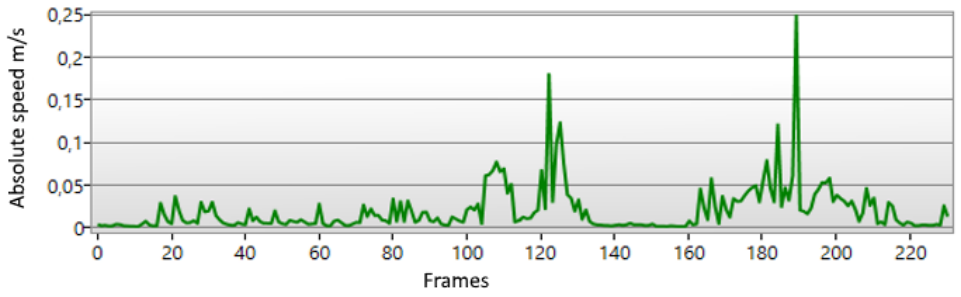

FIGURE 4.3: Absolute speed of the right ankle while performing actions on a sequence. It is possible to observe two moments in time where the absolute speed has increased, roughly [100, 140] and [160, 220]. These two moments refer to actions where the right ankle was moving significantly.

4.2.2 Absolute Speed-based Segmentation

We empirically observed that during the execution of each action some joints moved more than others. By measuring that movement and comparing it with other joints it would be possible to discern which joint is predominant in a certain action and assign a temporal segment to each joint. This information will be useful to perform the labeling of the temporal segments. Figure 4.4 shows a timeline which represents the movement of the right ankle. It is possible to discern two regions where that joint has a significant higher absolute speed. These two regions clearly depict moments in time where an action was performed that involved mainly the right leg.

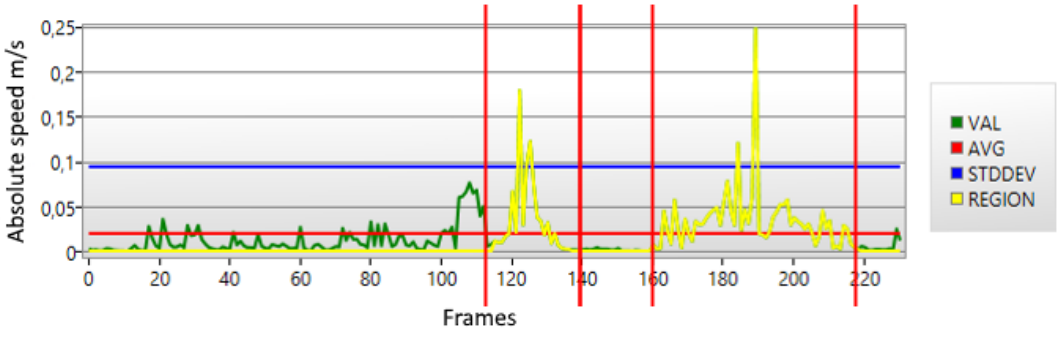

FIGURE 4.4: Regions of interest found by selecting frames in which the absolute speed of the moving joint was greater than the double of the standard deviation and above the average absolute speed. In this specific situation, two regions of interest were found, corresponding to two kicks.

Our first step was to create these regions which we denominated regions of interest. This was achieved by selecting frames in which the absolute speed value was above the standard deviation multiplied by a factor of two. We then selected all the neighboring frames that were above the average value with a tolerance of 3 frames below the average. This data was collected for four different joints: right and left ankle, right and left wrist. We then searched for overlapping regions. While the user performs a kick the rest of his body moves, specially the hands to maintain the body's balance. Overlapping regions were removed by considering only the joint moving at a higher average speed in each frame. Figure 4.5 illustrates an example result of our automatic segmentation method. Each color of the plot represents a temporal segment to which we assigned a joint as being the dominant joint for that action. We obtained 5 temporal segments which successfully correspond to the number of actions that the sequence contains, in this case: right-punch; left-punch; front-right-kick; front-left-kick; side-right-kick.

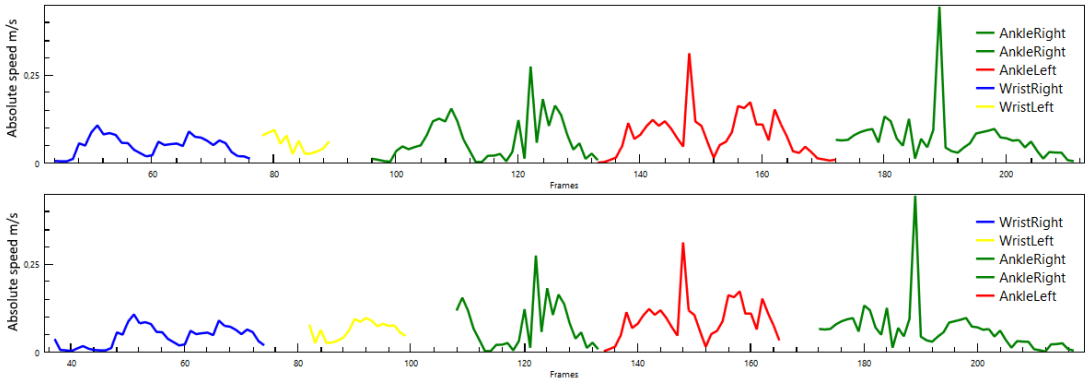

FIGURE 4.5: Absolute speed-based segmentation VS manual segmentation of a sequence. The upper chart illustrates the segmentation obtained by our absolute speed-based segmentation method. The bottom chart illustrates a segmentation based on the manual labeling of the frames of the sequence.

We applied the absolute speed-based segmentation method to all the 72 video sequences of our PRECOG dataset. Table 4.1 shows the average results per sequence for our absolute speed-based temporal segmentation. To measure the results we made a frame-by-frame comparison between corresponding temporal segments. Corresponding temporal segments are temporal segments which occur roughly at the same window-frame in the same sequence. An average segmentation accuracy of 83.04% is an interesting result. The difference in segmentation

accuracy between sequences can be explained by the actions that compose the sequences. Sequences 5 and 6 contain actions with a low intensity of movement (*back-fist and elbow-strike*). The movement of the remaining joints while performing those actions, is very similar to support the action being performed. This makes the task of temporal segmentation based on the absolute speed of the joints difficult to perform.

TABLE 4.1: Average temporal segmentation accuracy per sequence for our absolute speed-based temporal segmentation method. (Sequence description available in Table 3.2).

Sequence	1	2	3	4	5	6	Average
Segmentation Accuracy (%)	94.23	89.66	73.55	89.44	76.31	75.03	83.04

4.2.3 Warped K-means Segmentation

Warped K-Means (WKM) is a multi-purpose partitional clustering procedure that minimizes the Sum of Squared Errors criterion, while imposing a hard sequentiality constraint in the classification step (Leiva and Vidal, 2013). Action segmentation is a data partitioning problem, and can be addressed as a data clustering problem similar to those solved in Leiva and Vidal (2013) by WKM. This algorithm was, to the best of our knowledge, used to cluster trajectories derived from mouse and touch input and never used for the purpose of clustering human joint trajectories. Our intention is to apply WKM to the trajectories defined by the joints of the skeleton while the subject is performing an action. Figure 4.6 illustrates how WKM clusters an arbitrary shape.

For each sequence we had to process the data so that it could be used by the WKM algorithm. The first task was to automatically remove regions from the sequence where the subject was not moving. When we recorded the dataset some subjects did not execute the actions immediately, so in the beginning of most of the sequences there is a small region where the subject is stationary. This also happens at the end of the sequence were the person in charge of managing the capture would take some time to stop the recording after the subject has finished

FIGURE 4.6: Application of WKM clustering of an arbitrary shape. This shape could be hand drawn and the separation of segments would occur when the intensity of the motion decreased.

Source: Leiva and Vidal (2013)

performing the actions. An example of this is illustrated in Figure 4.7 with the highlighted regions.

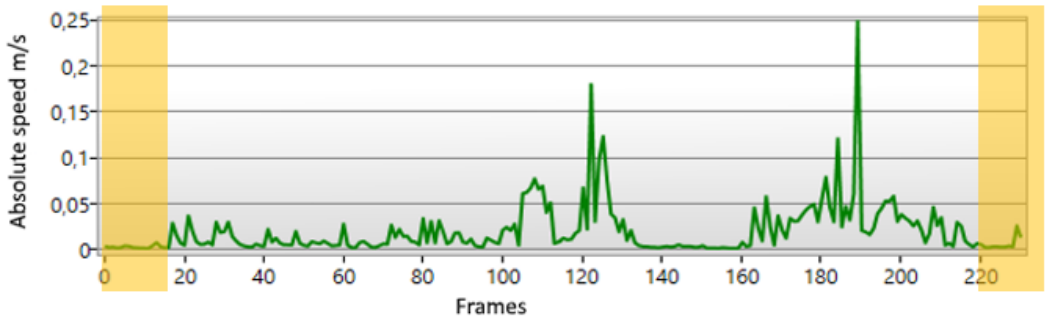

FIGURE 4.7: Highlighted regions of the sequence where the user is stationary. The absolute speed at those frames is almost zero.

In order to detect if the subject was moving we implemented a method to measure the average displacement of all the joints of the skeleton during a small time frame (five frames). If the average displacement is greater than a threshold defined from observation, we conclude that the subject is moving. This approach was successful in identifying stationary frames. For example sequence 1 performed by subject 1, from manually labeled data, action only begun at the frame 37 and ended at frame 211. Our method concluded that the subject initiated moving at frame 38(+1) and stopped at frame 207(-4). After removing the stationary frames we selected four joints (right and left ankle, right and left wrist) and calculated the average speed of these joints separately by referential axis. This would leave

us with 3 feature vectors representing the average speed of the four joints in X,Y and Z.

Each feature vector was saved as a column in a text file and then processed by WKM which will generate the following information for each sequence: *boundaries; clusters; centroids; local energy; total energy; iterations; number of transfers; cost.*. For a more detailed description of these features please refer to Leiva and Vidal (2013). The most relevant information is the boundaries which will be used to segment the sequence. A simplified explanation of the process is described by Algorithm 5.

Algorithm 5 Pseudo-code describing how we apply the Warped K-means method to perform temporal segmentation of all the sequences.

```
1: procedure WKM CLUSTERING(sequences)
2:     for all <sequences> do
3:         check for subject movement
4:         remove stationary frames
5:         data ← export speed per joint and per axis
6:         segments ← execute WKM(data)
7:     end for
8: end procedure
```

Figure 4.8 illustrates an example of WKM segmentation. The bottom-most row shows the ground-truth segmentation, top-most row is the segmentation obtained when performing WKM. We noticed that WKM has limitations in dealing with the frames between the actions or accurately detecting the ending of an action. The second action of the sequence (in red), the WKM temporal segment has nearly twice the length of the manual segmentation.

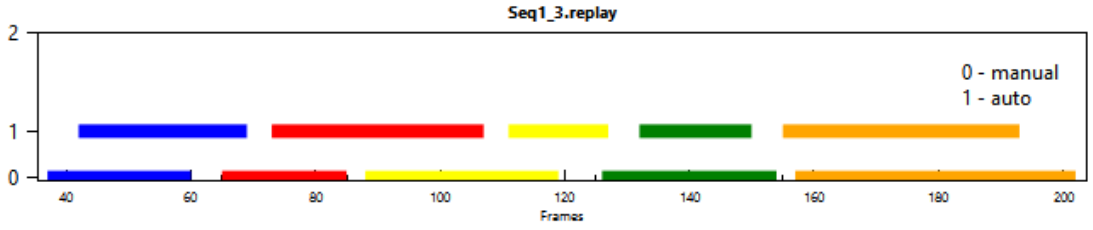

FIGURE 4.8: WKM temporal segmentation VS manual temporal segmentation. For this sequence in particular it is visible that the WKM temporal segmentation method introduces an offset in almost all the segments with some of them overlapping the following segments.

Table 4.2 shows the average segmentation accuracy obtained by WKM. In a frame-by-frame comparison the results are far from optimal. Since Warped K-Means (WKM) is a multi-purpose partitional clustering procedure we expected better results but unfortunately with an average segmentation accuracy of 55.23% this approach is not reliable enough to be used in our framework.

TABLE 4.2: Average temporal segmentation accuracy per sequence for our application of the Warped K-means temporal segmentation method to sequences of human activity.

Sequence	1	2	3	4	5	6	Average
Segmentation Accuracy (%)	63.55	49.74	40.86	45.20	45.17	59.02	55.23

4.2.4 Classifier-based Segmentation

Our previous segmentations methods fall in the category of *unsupervised* methods which do not require training data. Since the results were far from optimal we decided to experiment with a *supervised* method, where different kinds of models such as Hidden Markov Models (Bashir et al., 2007), Neural Networks (Hofmann and Buhmann, 1998) or Random Forests (Breiman, 2001) can be trained using manually segmented and labeled trajectories to perform segmentation of sequences of actions.

Figure 4.9 illustrates how a sequence is partitioned into the several actions that compose it (each color represents an action) then, highlighted in grey are the neighboring segments for each action which are not labeled to any action in particular. Since we proposed a method to perform action recognition in real-time on a frame-by-frame basis in our methodology, we decided to apply the same concept here and treat these neighboring segments as another action and train a classifier to recognize it. If the recognition of this action which we called *No-Action* revealed to be successful, then the sequence could be partitioned based on that information.

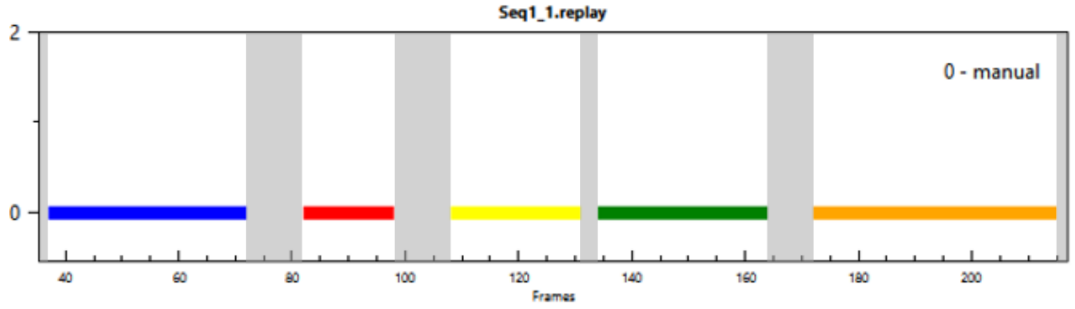

FIGURE 4.9: Manual temporal segmentation where each colored segment represents an action of the sequence and the grey segments represent the neighboring frames between actions.

In order to generate the training data we selected certain attributes that would be used as features. We selected speed in meters per second and orientation of four joints (right and left ankle, right and left wrist). Based on the position of the joints we calculated the speed of each joint (Eq. 4.3) instead of the absolute speed used in our absolute speed-based segmentation method for X,Y and Z. This calculation is performed in every 33 milliseconds which corresponds to the time between each frame captured by the Kinect sensor. Where the distance in meters per axis is again, calculated by the Equation 4.2.

$$s_i = \frac{d(J_i)}{\Delta t}, \; J_i = ith \; joint \; position \; in \; 3D \tag{4.3}$$

The bone orientation is provided in two forms:

- A hierarchical rotation based on a bone relationship defined on the skeleton joint structure.
- An absolute orientation in Kinect camera coordinates.

We experimented with both forms of rotation and although with a marginal difference, obtained better results with the absolute rotation of the bone. The orientation information is provided in form of quaternions and rotation matrices. Thus, the feature vector has 49 attributes:

- 3D speed vector for 4 joints, for XYZ=12.

- 3x3 orientation matrix for 4 bones = 36.
- action label = 1.

Since we want to perform action recognition for each frame of the sequence, each instance of the training data corresponds to a feature vector calculated from a frame of the sequence. We used a Random Forest of 100 trees to create a binary classifier model trained with a total of 142938 instances. In order to test the results we used k-fold cross-validation with the option of 10 folds. The training data has two classes: *No-Action* which corresponds to the neighboring frames of the actions, and *In-Action* which corresponds to any manually labeled action contained in the sequence. From 15882 instances, 93.14% were correctly classified, a more detailed description of the results can be found in Table 4.3.

TABLE 4.3: Detailed accuracy results showing precision, recall and f-measure in classifying the No-action and In-Action frames.

Class	Precision	Recall	F-Measure
No-Action	0.938	0.850	0.892
In-Action	0.928	0.972	0.950

Given an unknown sequence of actions, we ask the classifier to label each frame of the sequence as *No-Action* or *In-Action* and with that labeling we can easily obtain the classifier-based temporal segmentation (Figure 4.10).

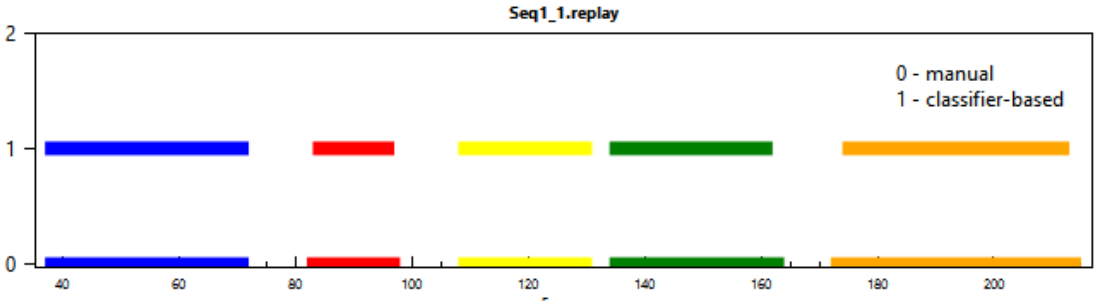

FIGURE 4.10: Classifier based temporal segmentation VS manual temporal segmentation. The significantly high segmentation accuracy of the classifier-based segmentation reflects the precision values above 90% in No-Action and In-Action classification.

4.3 Semi-unsupervised Labeling of Human Activity

In this section, we describe our experiments to perform semi-unsupervised labeling of human activity in RGB-D videos on two activity datasets (PRECOG and CAD-120). An action can be represented by the poses that the human body takes over time while performing an action. Each pose respects certain positions, degrees of freedom, constraints and orientation of joints of the skeleton. Our hypothesis is that with the correct features extracted from the skeleton tracker it is possible to cluster patterns that model the movements performed by the subjects.

4.3.1 Features

In order to limit the size of our feature vectors and based on related work (Nirjon et al., 2014) we compute the features described in Table 4.4 for the upper-skeleton joints (*left elbow, left wrist, right elbow and right wrist*) and also for the lower-skeleton joints (*left knee, left ankle, right knee, right ankle*). We experimented with more joints and the difference was marginal, even worst in some cases. Again we normalize the positions of the joints by redefining the 3D coordinates of all 20 skeleton joints using the spine joint as the frame of reference (Figure 4.11).

TABLE 4.4: These are the features that will be computed from the skeleton frames captured by the Kinect sensor and fed to a clustering algorithm.

Description	Count
absolute speed of each joint (4 joints)	4
velocity vector of each joint (4 joints)	12
accumulated displacement of each joint (4 joints)	12
flexion and extension angle between two bones	4
bone orientation (3x3 matrix)	36

The first feature that we computed was the absolute speed of each joint where absolute speed $= \left|\frac{distance}{\Delta t}\right|$ with $distance =$ final position $-$ initial position. As for the velocity vector, suppose the position of a joint at time t is given by the position

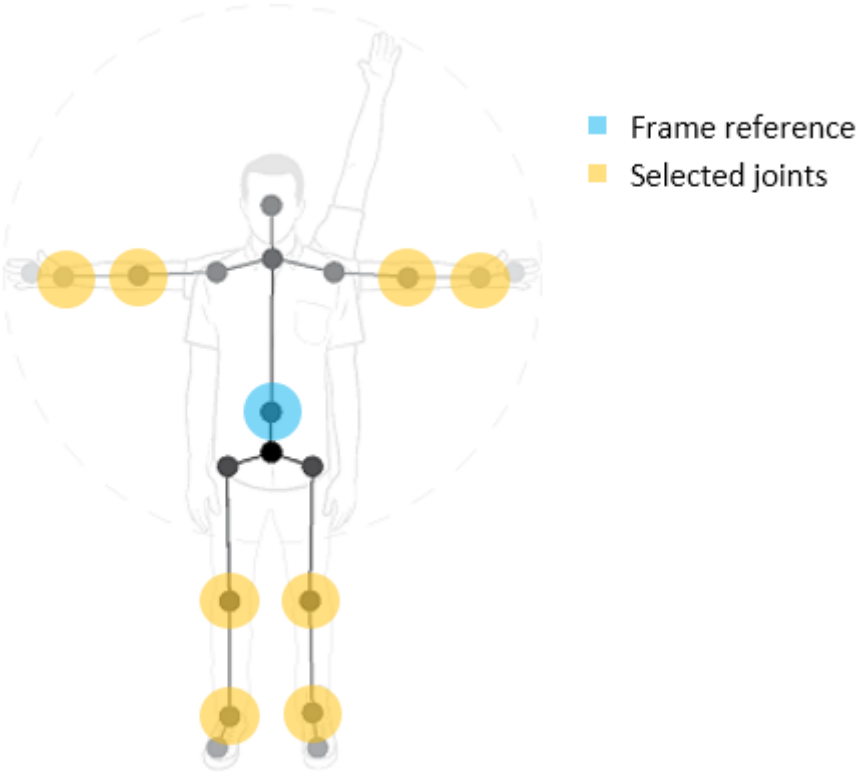

FIGURE 4.11: Illustration of the eight skeleton joints selected (in yellow) to extract features that will be used in the clustering of temporal segments. The hip joint (blue) will be used as the frame reference of the skeleton to normalize the positions of the joints.

Source: Kinect for Windows SDK 1.8 (2017)

vector $s(t) = (s_1(t), s_2(t), s_3(t))$. Then the velocity vector $v(t)$ is the derivative of the position, $v = \frac{(ds)}{(dt)} = (\frac{ds_1}{dt}, \frac{ds_2}{dt}, \frac{ds_3}{dt})$. The displacement vector in 3D quantifies both the distance and direction of the motion executed by a joint while performing an action. We also computed the flexion/extension angle of the elbows and the knees (Figure 4.12) using the law of cosines. Given the position of three joints in the 3D space (a = right shoulder, b = right elbow, c= right wrist) the angle γ is defined by Equation 4.4.

$$\gamma = \arccos\left(\frac{a^2 + b^2 - c^2}{2ab}\right), \text{for XYZ} \tag{4.4}$$

Experiments were made with different combinations of features to create the feature vector (Jardim et al., 2016c). The best results were obtained with the

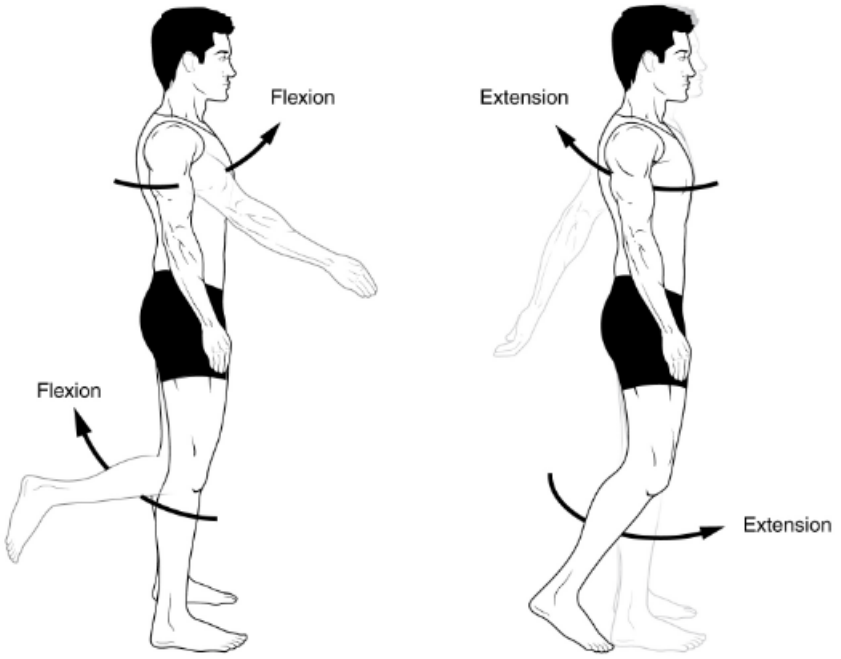

FIGURE 4.12: Flexion and extension describe bone movements that affect the angles between two bones of the body. Flexion decreases the angles between the bones and extension increases the angle between the bones. We calculated these angles for the elbow and knee joints to be used as features.

Source: Tonye Ogele CNX (2017)

following feature vector of $size = 20$:

- absolute speed (m/s) of each joint (4 joints), for XYZ = 4.
- normalized displacement in meters of each joint (4 joints), for XYZ = 12.
- flexion and extension angle between two bones (4 joints) = 4.

The clustering algorithm requires a fixed-length feature vector to be computed from each temporal segment found by our method. The sequences have variable length because different subjects take different amounts of time to perform the same actions, in the same way, the temporal segments found will also have variable length. Given the $N-length$ of a temporal segment $ts = (ts1, ts2, ..., tsN)$, where N is the window-size, we calculate the average value for each feature into a K-length sequence $k = (k1, k2, ..., kN)$ before it is sent to the clustering algorithm, where K is the size of the feature vector and it will vary depending on the combination of features used.

We tested two clustering algorithms: K-Means (Hartigan and Wong, 1979) and Hierarchical Clustering (Moore and Essa, 2002). K-means is a very well known algorithm and its aim is to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. Hierarchical Clustering is a method of cluster analysis which seeks to build a hierarchy of clusters, we adopted the agglomerative approach which is a *bottom up* approach: each observation begins in its own cluster, and pairs of clusters are combined as one moves up the hierarchy. We made various experiments and combinations with the features described above that were used to constitute the feature vectors for the clustering algorithm and obtained different results depending on the features used. In K-Means we used Euclidean distance as the distance function (which measures the distance between two individual instances) with a maximum of 500 iterations, for Hierarchical Clustering the distance function used was WARD, which finds the distance of the change caused by merging the cluster, again with a maximum of 500 iterations. In the initial experiments we tried to cluster all the temporal segments with information from all the joints at the same time but the performance was low (Table 4.5) with ambiguous results.

TABLE 4.5: Confusion matrix action-wise clustering results for hierarchical clustering algorithm applied on all the temporal segments found on all the sequences of the PRECOG dataset. With a total of eight distinct actions, where each cluster represents a different action without applying the body filtering method.

Action	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Right punch	0.0%	0.0%	8.16%	14.29%	0.0%	6.12%	69.36%	2.04%
Left punch	0.0%	3.03%	0.0%	78.79%	0.0 %	0.0%	18.18%	0.0%
Back fist	3.85%	15.38%	40.38%	13.46%	0.0%	0.0%	26.92%	0.0%
Elbow strike	0.0%	10.53%	52.63%	15.79%	0.0%	0.0%	21.05%	0.0%
Front right kick	14.29%	75.0%	1.79%	3.57%	0.0%	1.79%	3.57%	0.0%
Side right kick	33.75%	58.75%	0.0%	0.0%	3.75%	1.25%	1.25%	1.25%
Front left kick	0.0%	7.89%	0.0%	2.63%	7.89%	52.63%	0.0%	28.95%
Side left kick	5.13%	2.56%	2.56%	0.0%	26.92%	30.77%	3.85%	28.21%

Nevertheless we observed that different actions that were performed with the same body part were assigned to the same cluster. This information can be used to reduce the ambiguity of the data to be clustered and simplify the clustering process by clustering the temporal segments by body part and then by action. With this in mind, we implemented a method that divides the temporal segments in two groups: upper-body actions and lower-body actions. This is done based on

the most active joint that was assigned by the sampling method, for example, if the most active joint of a given segment is the right wrist, that temporal segment will be labeled as upper-body action. Table 4.6 shows the clustering results of the temporal segments contained in Sequence 1 using the K-means algorithm. Some of the temporal segments that refer to a *Side right kick* were labeled as a *Front right kick*, this is understandable since both actions are very similar and performed by the same body part.

TABLE 4.6: Confusion matrix action-wise clustering results for K-means clustering algorithm applied on the temporal segment found on Sequence 1. Each cluster represents a different action.

Action	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Right punch	91.67%	0.0%	0.0%	8.33%	0.0%
Left punch	0.0%	100.0%	0.0%	0.0%	0.0%
Front right kick	8.33%	0.0%	91.67%	0.0%	0.0%
Side right kick	0.0%	0.0%	33.33%	66.67%	0.0%
Side left kick	0.0%	0.0%	8.33%	8.33%	83.33%

Research from Kaur and Kaur (2013) refers that K-Means is usually more efficient in terms of its run-time, specially when dealing with large datasets, on the other hand Hierarchical Clustering, although slower in execution, has higher clustering performance. We confirmed these results, since our dataset is relatively small and we are performing clustering on sub-sets of identical sequences performed by different subjects, Hierarchical Clustering obtained better results in the tests (Table 4.7). Again, similar actions still represent a difficult task for the clustering process. For the sake of simplicity, we portray the comparison of these two clustering algorithms only for Sequence 1 of the dataset. From here on, all the clustering/labeling results were obtained via Hierarchical Clustering.

Table 4.8 shows the clustering results for all the sequences (high-level activities) and its sub-activities. The results are promising and consistently above 80% except for the *Elbow strike* action which is confused with the *Back fist* action (these two actions are very similar and performed by the same body part).

In order to verify the scalability of our approach we executed the same process described in Algorithm 1 using the CAD-120 dataset, which proved to be a

TABLE 4.7: Confusion matrix action-wise clustering results for hierarchical clustering algorithm applied on the temporal segment found on Sequence 1. Each cluster represents a different action.

Action	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Right punch	100.0%	0.0%	0.0%	0.0%	0.0%
Left punch	0.0%	100.0%	0.0%	0.0%	0.0%
Front right kick	0.0%	0.0%	75.0%	0.0%	25.0%
Side left kick	0.0%	0.0%	0.0%	100.0%	0.0%
Side right kick	0.0%	0.0%	16.67%	0.0%	83.33%

TABLE 4.8: Confusion matrix action-wise clustering results for hierarchical clustering algorithm applied on all the temporal segments found on all the sequences of the PRECOG dataset. With a total of eight distinct actions, where each cluster represents a different action. Here the body filtering method is applied to distinguish upper-body actions from lower-body actions and only then we apply the clustering of actions.

Action	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Right punch	89.58%	2.08%	8.33%	0.0%	0.0%	0.0%	0.0%	0.0%
Left punch	4.17%	95.83%	0.0%	0.0%	0.0 %	0.0%	0.0%	0.0%
Back fist	13.89%	0.0%	80.56%	5.56%	0.0%	0.0%	0.0%	0.0%
Elbow strike	0.0%	0.0%	33.33%	66.67%	0.0%	0.0%	0.0%	0.0%
Front right kick	0.0%	0.0%	0.0%	0.0%	83.33%	14.58%	0.0%	2.08%
Side right kick	0.0%	0.0%	0.0%	0.0%	16.67%	80.0%	0.0%	3.33%
Front left kick	0.0%	0.0%	0.0%	0.0%	0.0%	2.78%	83.33%	13.89%
Side left kick	0.0%	0.0%	0.0%	0.0%	0.0%	4.17%	5.56%	90.28%

challenging dataset for semi-unsupervised labeling. The presence of a left-handed subject makes it very difficult for the clustering algorithm to group similar actions if they are performed with different joints. The results obtained were far from optimal (Table 4.9). The actions in this dataset are very distinct from the actions of our dataset. Some of the labellings make sense, *Opening* and *Closing* as *Eating* and *Drinking* are very similar and using only the skeleton information is very difficult to distinguish them. To mitigate these ambiguities Koppula et al. (2013) added scene/object recognition and tracking to aid the process of learning human activities.

Given the amount of incorrectly clustered instances our hypothesis is that if we train the classifiers with this labeling the classifiers will be induced to error and the model will fail to recognize accurately the actions in real-time. Our expectation is that models that deal well with noise could correct some of the labeling problems

TABLE 4.9: Confusion matrix action-wise clustering results for hierarchical clustering algorithm applied on all the temporal segments found on all the sequences of the CAD-120 dataset. With a total of eight distinct actions, where each cluster represents a different action.

Action	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Reaching	12.10%	42.74%	0.0%	6.85%	17.74%	0.0%	5.24%	15.32%
Moving	10.31%	14.38%	1.88%	29.38%	28.13 %	0.0%	4.06%	11.98%
Pouring	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%
Eating	0.0%	0.0%	0.0%	62.50%	0.0%	0.0%	0.0%	37.50%
Drinking	0.0%	0.0%	0.0%	69.23%	0.0%	0.0%	0.0%	30.77%
Opening	0.0%	0.0%	71.15%	0.0%	1.92%	1.92%	0.0%	25.0%
Placing	15.69%	0.0%	23.53%	9.15%	0.65%	34.64%	8.50%	7.84%
Closing	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%

by generalizing well. This lead us to essay a few experiments based on this data, if nothing else, as a baseline for other results. Experiments to validate this hypothesis are shown in the next section.

4.4 Human Activity Recognition

This section describes the experimental results of our temporal segment and frame-by-frame action recognition approaches. We test our model on the PRECOG and the CAD-120 datasets. In order to train the classifiers we have to generate the training data. We have two versions of each dataset: manually labeled (frame-by-frame) and automatically labeled (temporal segment). For the automatically labeled dataset we created a procedure that, given the label of the temporal segment, applies the same label to all the frames within that segment, resulting in a dataset labeled frame-by-frame. To verify our hypothesis, we experimented with several supervised learning models: MLP (Rumelhart et al., 1986; Kubat, 1999); SVM using pairwise classification (Platt, 1998) and RF which are a combination of tree predictors (Breiman, 2001). Using distinct classifiers allowed us to verify if different machine learning methods obtained different results and also compare the performance of our action recognition framework trained with data labeled in a semi-unsupervised way, versus data manually labeled.

4.4.1 Features

For each labeled frame we compute an activity feature vector which is given as input to the classifier together with the label of the action. The *wrist-right; wrist-left; ankle-right; ankle-left* joints were selected to compute the velocity vector and the *elbow-right; elbow-left; knee-right; knee-left;* joints were selected to extract the bone orientation as shown in Figure 4.13.

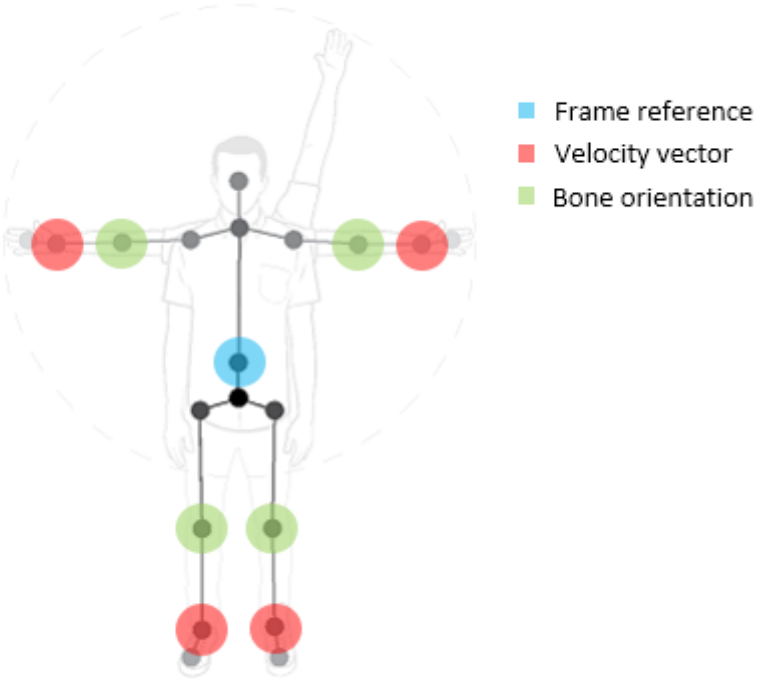

FIGURE 4.13: Illustration of the eight skeleton joints selected to extract features that will be used to generate the activity feature vector. The hip joint (blue) will be used as the frame reference of the skeleton to normalize the positions of the joints.

Source: Kinect for Windows SDK 1.8 (2017)

The 3D coordinates are with respect to the hip joint which will be used as a frame of reference centered at the Kinect sensor point of view. Frames from the camera are converted into feature vectors which are invariant to relative position and orientation of the body since we normalized their positions. The computed activity feature vector is comprised of the following features with $size = 49$:

- velocity vector [X, Y, Z] of each joint, J_i (4 joints) = 12.

- bone orientation obtained from a quaternion (3x3 matrix)(4 joints) = 36.
- label of the action = 1.

The bone orientation is provided by the Kinect SDK which returns a *BoneRotation* object that has a rotation matrix and a quaternion vector (Figure 4.14). To calculate the absolute orientation of each bone, we have to multiply the rotation matrix of the bone by the rotation matrices of the parents (up to the root joint).

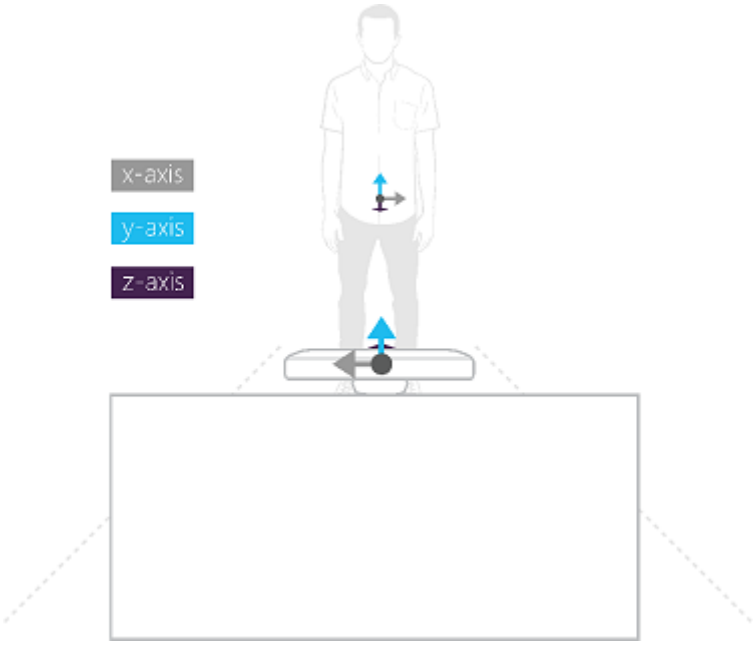

FIGURE 4.14: The absolute subject rotation in camera space coordinates is provided by the hip center joint. This means that the subject object space is centered at the hip center joint. The x axis is horizontal, the y axis is vertical and the z axis refers to the depth.

Source: Kinect for Windows SDK 1.8 (2017)

4.4.2 Temporal Segment Recognition

To evaluate our temporal segment recognition approach k-fold cross-validation will be used to randomly split the data in the training set into k smaller sets with a single sub-set being left out as a test-set with k=*10*. The results obtained in the tables below represent the average recognition accuracy values of 30 trials using

random seed values with the corresponding standard deviation to quantify the amount of variation in performance that occurred in each trial.

Table 4.10 shows the average recognition accuracy obtained for three distinct multi-class classifiers (MLP, SVM and RF) trained with all the actions from the manually labeled training set and the results are significantly below the state-of-the-art (Nirjon et al., 2014). There is a clear difference in performance between the ML methods. RF has nearly 10% increase in performance compared to MLP and SVM.

TABLE 4.10: Temporal segment classification accuracy (%) using multi-class classifiers trained with semi-unsupervised labeled data and corresponding standard deviation between trials for the PRECOG dataset.

Action	MLP	SVM	RF
right-punch	69.80 ±0.83%	72.08 ±0.17%	80.89 ±1.00%
left-punch	70.22 ±0.77%	72.03 ±0.21%	81.30 ±1.28%
front-right-kick	70.11 ±1.03%	72.01 ±0.20%	81.37 ±0.88%
front-left-kick	69.99 ±0.92%	72.11 ±0.17%	81.39 ±1.11%
side-right-kick	69.97 ±0.72%	72.07 ±0.19%	81.57 ±0.83%
side-left-kick	70.10 ±0.87%	72.10 ±0.17%	81.54 ±1.67%
backfist	69.88 ±0.80%	72.10 ±0.17%	80.97 ±0.81%
elbow-strike	70.04 ±0.75%	72.04 ±0.20%	81.12 ±0.80%

Based on the binary classifiers approach from (Nirjon et al., 2014) we trained eight binary supervised classifiers using manually labeled data for recognizing the eight aggressive actions contained in our dataset. Each classifier is trained to distinguish one action from all others. This approach (binary classifiers) produced the best results in Nirjon et al. (2014) using SVM classifiers. Table 4.11 clearly shows the difference between using binary-classifiers versus multi-class classifiers with the obvious advantage of using binary classifiers with average accuracies above 91% in recognizing an action surpassing the average results from Nirjon et al. (2014) of 90%. In this case the difference in accuracy between classifiers is reduced, but again RF manages to obtain the best results.

We also present a detailed accuracy by class in Table 4.12 referring only to the RF classifier. Even with these results is important to outline that a multi-class

TABLE 4.11: Temporal segment classification accuracy (%) using multi-class classifiers trained with manually labeled data and corresponding standard deviation between trials for the PRECOG dataset.

Action	MLP	SVM	RF
right-punch	94.24 ±0.44%	91.52 ±0.17%	90.08 ±0.37%
left-punch	89.09 ±0.44%	92.50 ±0.26%	92.21 ±0.37%
front-right-kick	88.14 ±0.96%	87.95 ±0.21%	93.20 ±0.53%
front-left-kick	89.96 ±0.79%	90.42 ±0.28%	91.97 ±0.48%
side-right-kick	91.22 ±0.16%	91.92 ±0.07%	94.53 ±0.57%
side-left-kick	83.62 ±0.97%	84.76 ±0.23%	91.74 ±0.51%
backfist	92.55 ±0.32%	92.77 ±0.00%	93.58 ±0.46%
elbow-strike	95.02 ±0.28%	96.66 ±0.00%	96.66 ±0.00%

classifier has the advantage of always being able to provide a classification for a given instance.

TABLE 4.12: Temporal segment recognition results on our PRECOG dataset showing precision, recall and f-measure for action recognition of the binary classifiers using manually labeled data with random forests algorithm.

Precision	Recall	F-Measure	Class
0.923	0.902	0.912	right-punch
0.943	0.702	0.805	left-punch
0.945	0.809	0.872	front-right-kick
0.953	0.888	0.919	front-left-kick
0.872	0.812	0.841	side-right-kick
0.929	0.872	0.899	side-left-kick
0.919	0.861	0.889	backfist
0.993	0.552	0.709	elbow-strike

Table 4.13 shows the results of the repetition of the previous experiment with the fundamental difference of using a training set labeled by our semi-unsupervised labeling pipeline. As before, the MLP classifier has the worst performance and RF performs the best. MLP in comparison to the values of Table 4.11 has the largest decrease in performance, in some cases more than 10%. Concerning SVM and RF the difference is much less, never surpassing 3%.

As expected, the usage of semi-unsupervised labeling affects the accuracy of the classifiers. This can be explained by the error that our automatic labeling method introduces. This error is caused by frames that might be added to the

TABLE 4.13: Temporal segment classification accuracy (%) using multi-class classifiers trained with semi-unsupervised labeled data and corresponding standard deviation between trials for the PRECOG dataset.

Action	MLP	SVM	RF
right-punch	83.82 ±0.81%	88.29 ±0.16%	89.40 ±0.48%
left-punch	82.43 ±1.31%	90.20 ±0.00%	90.84 ±0.33%
front-right-kick	81.22 ±0.74%	90.75 ±0.07%	90.00 ±0.49%
front-left-kick	89.99 ±0.76%	87.91 ±0.13%	90.99 ±0.25%
side-right-kick	82.80 ±1.18%	87.88 ±0.07%	89.57 ±0.57%
side-left-kick	84.99 ±0.86%	90.28 ±0.05%	90.56 ±0.68%
backfist	83.09 ±1.44%	87.60 ±0.00%	90.05 ±0.41%
elbow-strike	95.90 ±0.31%	96.83 ±0.00%	96.83 ±0.00%

segment where in fact they do not belong to the action, or the opposite, frames that belong to the action are left out of the segment. Another source for error is our clustering and labeling method which confuses similar actions that lead to an incorrect labeling. Nonetheless, the difference is relatively small, and depending on the application, it could be negligible and remove the necessity of having to rely on human resources to manually label data. Again, we present a detailed accuracy by class in Table 4.14 referring only to the RF classifier.

TABLE 4.14: Temporal segment recognition results on our PRECOG dataset showing precision, recall and f-measure for action recognition of the binary classifiers using semi-unsupervised labeled data with random forests algorithm.

Precision	Recall	F-Measure	Class
0.872	0.921	0.896	right-punch
0.924	0.923	0.924	left-punch
0.876	0.951	0.912	front-right-kick
0.903	0.916	0.910	front-left-kick
0.891	0.919	0.905	side-right-kick
0.883	0.936	0.909	side-left-kick
0.898	0.928	0.913	backfist
0.931	0.976	0.953	elbow-strike

Finally, in Table 4.15 we calculate the difference in performance for each classifier accuracy using manually labeled data and semi-unsupervised labeled data. In some cases, the classifier that was trained using semi-unsupervised labeled data outperformed its counterpart.

TABLE 4.15: Difference in classification accuracy (%) between models that were trained with manually labeled data versus semi-unsupervised labeled data for each binary classifier per action. Negative values represent the loss in accuracy of the models trained with the semi-unsupervised data.

Action	MLP	SVM	RF
right-punch	-10.42 %	-3.23 %	-0.68 %
left-punch	-6.66 %	-2.30 %	-1.37 %
front-right-kick	-6.92 %	2.80 %	-3.2 %
front-left-kick	0.03 %	-2.51 %	-0.98 %
side-right-kick	-8.42 %	-4.04 %	-4.96 %
side-left-kick	1.37 %	5.52 %	-1.18 %
backfist	-9.46 %	-5.17 %	-3.53 %
elbow-strike	0.88 %	0.17 %	0.17 %
average	-4.95 %	-1.09 %	-1.97 %

All the results presented here were obtained *off-line* in WEKA. When we tried to replicate the results in *real-time*, we noticed the shortcoming of this approach. The issue with temporal segment recognition is that several frames are required to compute an instance of the feature vector. How many frames have to be captured to compute the feature vector? Actions have varying length and duration. If the number of frames is reduced, their average value might not be representative of an action, but if we use too many the recognition will occur with a delay of n-frames and it will not be useful. In summary this approach might be useful, but only for *off-line* classification because it requires access to the whole sequence of actions.

4.4.3 Frame-by-frame Recognition

The frame-by-frame recognition uses the same activity feature vector as the temporal segment recognition approach. Instead of computing the activity feature vector for a set of frames, we compute a new activity feature vector for every frame captured by the Kinect sensor. This approach will be evaluated using 10-fold cross validation with random forest classifier and the global performance is given by the average precision, recall and f-measure. High accuracy results were obtained with this approach for the PRECOG dataset using manually labeled data (Table 4.16), where the average precision and recall are 97.3% and 98.3%, respectively.

TABLE 4.16: Frame-by-frame recognition results on our PRECOG dataset showing precision, recall and f-measure for action recognition of the binary classifiers using manually labeled data with random forests algorithm.

Action	Correct	Incorrect	Precision	Recall	F-Measure
right-punch	97.45%	2.55%	0.969	0.976	0.949
left-punch	98.34%	1.66%	0.981	0.984	0.983
front-right-kick	97.53%	2.47%	0.974	0.983	0.978
front-left-kick	98.29%	1.71%	0.981	0.985	0.983
side-right-kick	96.35%	3.65%	0.959	0.977	0.968
side-left-kick	97.84%	2.16%	0.975	0.985	0.980
backfist	97.17%	2.83%	0.964	0.989	0.976
elbow-strike	97.65%	2.35%	0.984	0.983	0.983

Our results were significantly higher than the results presented in Nirjon et al. (2014) of an action recognition accuracy of 90%, obtained using binary classifiers also. The results are in-line with more recent approaches (Gaglio et al., 2015; Cippitelli et al., 2016) using datasets with similar actions. Our approach has an advantage on performance: while Gaglio et al. (2015) performs the recognition of a sequence (i.e., posture analysis and activity recognition) in about 1 second, we perform the same task for every captured frame from the Kinect sensor (30 frames/s).

TABLE 4.17: Frame-by-frame recognition results on our PRECOG dataset showing precision, recall and f-measure for action recognition of the binary classifiers using semi-unsupervised labeled data with random forests algorithm.

Action	Correct	Incorrect	Precision	Recall	F-Measure
right-punch	88.90%	11.10%	0.867	0.887	0.877
left-punch	94.59%	5.41%	0.951	0.934	0.942
front-right-kick	80.65%	19.35%	0.799	0.831	0.810
front-left-kick	80.80%	19.20%	0.807	0.814	0.811
side-right-kick	78.10%	21.90%	0.787	0.811	0.790
side-left-kick	88.86%	11.14%	0.888	0.905	0.896
backfist	78.85%	21.15%	0.770	0.828	0.798
elbow-strike	75.00%	25.00%	0.714	0.837	0.771

We repeated the same experiments using data automatically labeled to train the binary classifiers (4.17). Again and as expected there was a decrease in recognition accuracy. The clustering results from Table 4.8 imply that there will be data mislabeled, this will add noise to the training set and induce the classifiers to

error leading to a decrease of the average precision and recall to 82.3% and 85.6%, respectively. The action with the highest accuracy is the *left-punch*, very likely because it is the only action executed mainly by the left arm of the subject.

We also evaluate the performance of our approach on the CAD-120 dataset. In this dataset the limitations of Kinect's tracking algorithm are more evident. Almost in every sequence and due to partial occlusions, one or more joints were mis-detected resulting in unnatural poses. When this occurs the skeleton tracker tries to infer the joints position according to the global skeleton. If a joint is not clearly visible it makes the whole recognition process more difficult and unreliable. To mitigate this we implemented a simple linear interpolation method that will interpolate the position of the joints when the tracker is unable to infer them. The interpolation is done for all the frames in between the last known position and the next known position. Figure 4.15 illustrates an example of interpolated joints versus not tracked joints.

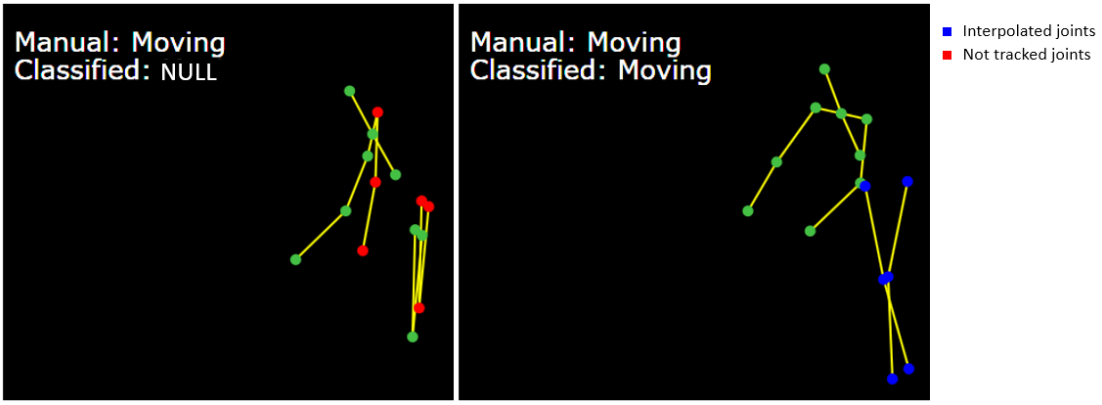

FIGURE 4.15: Real-time output of the action recognition application being performed on the CAD-120 dataset. (a) When occlusion occurs the Kinect sensor loses track of the joints which then are represented in red. (b) Interpolated joints (blue) are computed and replace the not tracked joints.

The evaluation setting proposed is the same as with our dataset, 10-fold cross-validation which means that the system is trained on 90% of the data and that 10% is used for testing. For this dataset, since the skeleton is also captured using Microsoft SDK, we could easily compute the same features that we computed for our dataset.

TABLE 4.18: Frame-by-frame recognition results on the CAD-120 dataset showing precision, recall and f-measure for action recognition of the binary classifiers using manually labeled data with random forests algorithm.

Action	Correct	Incorrect	Precision	Recall	F-Measure
reaching	96.89%	3.11%	0.952	0.953	0.952
moving	100.00%	0.00%	1.000	1.000	1.000
pouring	99.99%	0.01%	1.000	1.000	1.000
eating	100.00%	0.00%	1.000	1.000	1.000
drinking	99.59%	0.41%	0.997	0.993	0.995
opening	100.00%	0.00%	1.000	1.000	1.000
placing	96.81%	3.19%	0.958	0.943	0.950
closing	99.99%	0.01%	1.000	1.000	1.000

The highest performance for the proposed algorithm was obtained with the RF classifier, and the performance in terms of precision and recall, for each activity, is shown in Table 4.18. The recognition performance achieved with this dataset surpasses the results obtained by Koppula and Saxena (2016) in recognizing sub-activities in the same dataset. We obtained similar recognition performance with the work of Cippitelli et al. (2016), although a direct comparison cannot be made since they used a different dataset (KARD dataset from Gaglio et al. (2015)). Our hypothesis is that, one of the reasons for this result is the amount of data generated for training, since we use every frame to create an instance of the feature vector we have for example 535823 instances for the *moving* sub-activity which is a considerable amount of training data for a single action.

Our semi-unsupervised labeling method did not scale well for this dataset. Nevertheless we trained the classifiers with the automatically labeled data. The results are presented in Table 4.19 and the low precision obtained clearly illustrates that correctly labeled training data is very important to obtain relevant results.

Finally, Figure 4.16 illustrates a snapshot of our application performing real-time classification in the PRECOG dataset. The label *Classified* informs that the recognized action at the current frame from the Kinect sensor is a *side-left-kick*, which matches the *Manual* label which is taken from the manually labeled data.

TABLE 4.19: Frame-by-frame recognition results on the CAD-120 dataset showing precision, recall and f-measure for action recognition of the binary classifiers using semi-unsupervised labeled data with random forests algorithm.

Action	Correct	Incorrect	Precision	Recall	F-Measure
reaching	66.74%	33.26%	0.636	0.529	0.578
moving	68.52%	31.48%	0.697	0.773	0.733
pouring	61.00%	39.00%	0.612	0.614	0.613
eating	56.99%	43.01%	0.565	0.589	0.577
drinking	65.95%	34.05%	0.624	0.563	0.592
opening	65.43%	34.57%	0.630	0.584	0.606
placing	60.31%	39.69%	0.583	0.505	0.541
closing	66.99%	33.01%	0.683	0.747	0.714

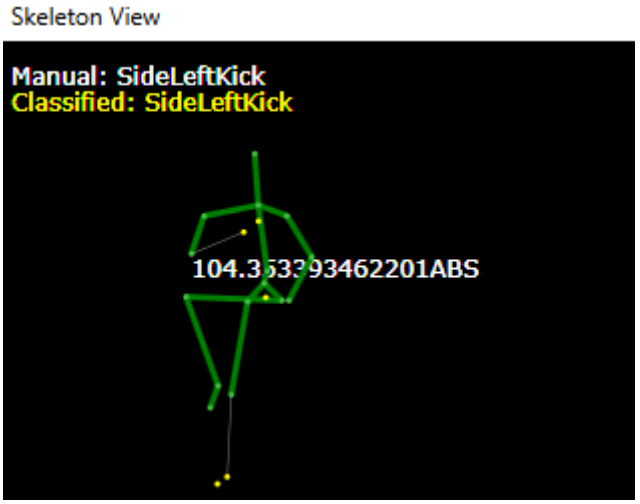

FIGURE 4.16: Illustration of our PRECOG application performing real-time action recognition. It displays the skeleton tracked by the Kinect sensor, the ground-truth labeling and the classified action.

4.5 Discussion

In this chapter we explored: (i) temporal segmentation of human activity, (ii) a semi-unsupervised labeling method of RGB-D videos which models skeleton movement using clustering algorithms and (iii) the problem of real-time human activity recognition using semi-unsupervised and manually labeled training data with extensive analysis of the proposed approaches on two datasets.

We experimented with several methods for performing temporal segmentation of human activity. Completely unsupervised human activity segmentation is a very challenging problem, even the state-of-the-art approaches have limitations

with applicability confined to short sequences of actions. Approaches which are based on thresholds of movements reveal to be highly dependent on the data and do not scale well when applied directly to another dataset. Our best results were obtained with a supervised method by learning the optimal labeling from multiple temporal segmentation hypotheses of various sequences. This approach has the limitation of being unable to deal with new actions. These results helped us to implement the first module (temporal segmentation) of our framework whose ultimate goal is to recognize and predict human activity in a sequence of actions.

We considered a semi-unsupervised method to label sequences of actions comprised by several sub-activities performed over long periods of time recorded with the Kinect sensor. We formulated the labeling/annotation problem as a clustering problem, and showed that a temporal segmentation and clustering algorithm can be used to label identical sub-activities performed by different users, thus reducing the amount of input required from a human judge to label a human activity dataset. Results indicate that ambiguous actions are difficult to label and improvements to the semi-unsupervised labeling/annotation method are required to guarantee the scalability of the approach to different datasets with ambiguous actions.

Extensive experiments for real-time action recognition were performed with comparisons between several supervised classifiers used on the state-of-the-art to recognize human activity. The two different methods that we proposed to perform real-time action recognition (temporal segment recognition and frame-by-frame recognition) were tested with manually labeled and semi-unsupervised labeled training sets. We showed that our frame-by-frame recognition method using manually labeled data surpasses the state-of-the-art approaches for similar actions. The usage of semi-unsupervised labeled data for training resulted in a decrease in performance due to the error that our labeling method introduces. The limitation of these type of supervised approaches is the inability to deal with new actions. In a real life scenario it is possible to have captured frames that will not belong to any of the trained actions, the classifier will simply output the class which has the higher probability even if it represents a different action.

After exploring in depth the action recognition problem we believe that the next logical step is to address the prediction of human activity. In Chapter 5 we present our proposal to model the relation of actions and sequences of actions, which can be used to predict the future actions based on the recognition of the past actions.

Chapter 5

Early Recognition and Prediction of Human Activity

"Education is the most powerful weapon which you can use to change the world."

Nelson Mandela

Our goal is to predict what a human subject will do next based on the current action that he is performing and the history of actions that were performed. Given the history of actions performed by a human for time t in the past and the current action a being performed, we intend to predict the next possible action a'. For example, if a human subject has picked a glass of water, what will be the outcome? Will he drink the water? Or will he spill it on the sink?

In this chapter, we demonstrate our approach for early recognition of human activity and how we address the problem of real-time human activity prediction using *n-grams* and conditional random fields. This chapter is structured as follows: in Section 5.1 we describe the experimental setup; in Section 5.2 we demonstrate our early recognition approach; in Section 5.3 we show how *n-grams* can be used to perform action prediction; in Section 5.4 we perform real-time human activity prediction using conditional random fields; and in Section 5.5 we discuss the results.

5.1 Experimental Setup

The performance of our solution for early recognition and prediction of human activity will be evaluated on two public 3D human activity datasets: PRECOG dataset and the CAD-120 dataset. The classification methods will be evaluated by using k-fold cross-validation. A combination of the Kinect for Windows SDK, WEKA and CRFSharp [1] which is a Conditional Random Fields library implemented in .NET(C#) was used to implement and conduct the experiments.

5.2 Early Recognition

The purpose of early recognition is to recognize unfinished activities as opposed to the after-the-fact classification of completed activities (Ryoo, 2011). Early recognition has been approached by Ryoo (2011); Mainprice and Berenson (2013); Hoai and De La Torre (2014) amongst others in the past and in most cases they were able to correctly classify ongoing activities, even when less of the first half of video containing the activity was provided. Early recognition of human activity is important to our approach because it will allow us to build a responsive and proactive system, where the classification is done as soon as possible. Figure 5.1 shows sequential frames from a temporal segment that correspond to the same action. Ideally all these frames should be labeled as the same action, independently of where they are located in the temporal segment. We present a very simple approach which is directly related to the design of our recognition framework. Since we proposed a real-time recognition method which is performed on every captured frame, we observed that our system was able to frequently recognize the current action right from the very first frames of the temporal segment.

In order to test the ability of our approach to recognize ongoing activities at an early stage, we measured the system recognition performances at the beginning of each temporal segment which represents a sub-activity. Specifically we select the

[1] https://github.com/zhongkaifu/CRFSharp

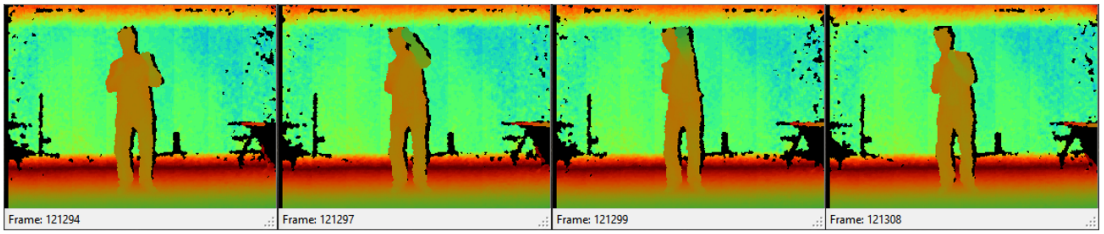

FIGURE 5.1: Illustration of a sample of depth frames (non-continuous) that were captured while the user was performing a *right-punch*.

first five frames of each temporal segment. Figure 5.2 describe the classification accuracy for the first five frames of the activities. With accuracy values between 85% and 98% these experimental results confirm that the proposed approach is able to correctly recognize this type of short-span human activities even at their earlier stage.

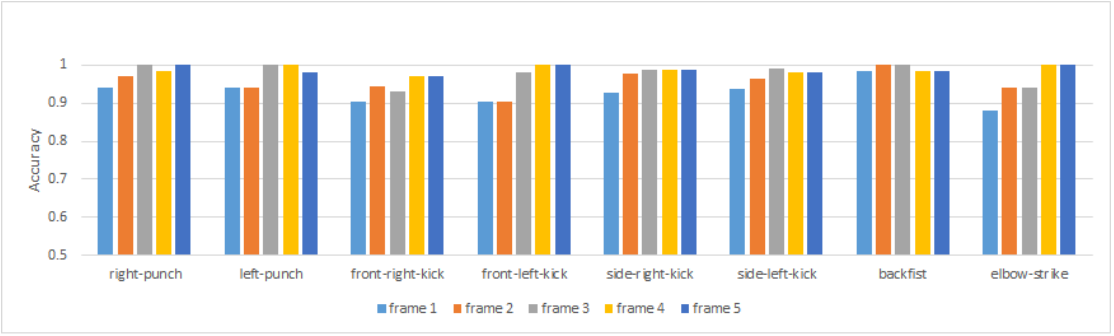

FIGURE 5.2: Frame-by-frame classification result of the first five frames of every temporal segment (which represents an action) contained in the test data.

During the experiments for early recognition we identified another interesting feature of our system: the ability to perform early recognition of the sub-activity which surpasses the ground-truth classification. Take the example of Figure 5.3, which illustrates a snapshot of the application recognizing a specific frame from the Kinect sensor that was manually labeled as *None* but in fact it should have been labeled as *side-right-kick*. Since the dataset was manually labeled by a human, the labeling might be incorrect in some frames. This might occur on the extremities of the temporal segment where these frames could be left unlabeled as *None*. These results, although very interesting, are a *by-product* of our real-time frame-by-frame action recognition. This is a reinforcement of the quality of our action recognition

method, which shows the ability to generalize and correctly classify instances that might not have been identified with the correct label even by a human judge.

FIGURE 5.3: Screenshot of the PRECOG application performing early action recognition. From the image, it is possible to observe that a correct classification of the action by the classifier was done, even when the ground-truth frame was not manually labeled as an action frame.

5.3 N-Gram Action Prediction

To perform action prediction with n-grams, we have to recognize the actions being performed by a human subject and store them in a list of past actions. In this specific case an n-gram is defined as a contiguous sequence of n actions, therefore this method requires knowledge of at least the two previous actions performed. The second column of Table 5.1 shows the probability of accurately predicting the next action using the Bayes theorem with noiseless data. The following columns show the performance of our approach using several classifiers trained with semi-unsupervised labeled data. The results are very similar between columns 3-5 and, as expected, the accuracy improves as we add more actions as input. Compared with column 2, we notice a loss of performance. Since our frame-by-frame action recognition method does not have a perfect accuracy of 100%, the training data generated by a classifier for this prediction method might have some mislabeled actions. For example, we might have the following labels assigned to a sequence of

actions as follows: "*right-punch, left-punch, NONE, side-left-kick, front-left-kick*". NONE represents a situation where the activity recognition module was unable to assign a label to the observed action. The third action of the sequence labeled as NONE would affect negatively the training. This means that the prediction accuracy will be highly dependent on the ability of the system to correctly recognize the actions performed by the subject to build the correct n-grams that were used to train the models. When the whole sequence is known (n-gram = 5), the classifiers converge identically and obtain the same prediction accuracy.

TABLE 5.1: Prediction accuracy comparison for the next action between different classifiers and with an increasing number of actions (*n-grams*) as input.

n-gram	Ground-Truth	SMO	RF	MLP
3	83.3%	77.7%	79.2%	77.7%
4	91.7%	86.8%	79.2%	89.6%
5	100%	95.8%	95.8%	95.8%

5.4 Conditional Random Fields Action Prediction

The proposed method applies conditional random fields to obtain a distribution of the future possible actions that will occur by sampling sequences of actions. We see fit the application of CRF to this problem due to their sequence modeling nature and structured prediction ability in a Part-of-Speech Tagging (POS) fashion as in NLP problems. Constructed on top of the frame-by-frame action recognition module from Chapter 4, we will group the classified frames into temporal segments that represent the current action that it is being performed, and for every recognized action store it in a list representing the history of actions. The pair action and history of actions comprises the feature vector. Every sequence of the PRECOG and the CAD-120 dataset was sampled into a list of pairs that include possible actions and the corresponding history of actions. This data will be used to train the models and perform the experiments, validated by using 10-fold cross validation. For each fold, the data is split 90% for training and 10% for testing.

The CAD-120 dataset is larger than the PRECOG and the sequences are composed by more sub-activities thus, the amount of data generated for training the CRF classifier is greater (Table 5.2).

TABLE 5.2: Training and testing corpus used for the PRECOG and the CAD-120 dataset. The 72 sequences of the dataset were sampled into 722 actions combinations which then were randomly split into training and testing.

	Training	Testing
PRECOG sequences	65	7
CAD-120 sequences	100	12
PRECOG actions combinations	650	72
CAD-120 actions combinations	5766	640

From Algorithm 4 when the recognition module classifies a new action we ask what is the most likely action that will occur next? We used manually labeled data and semi-unsupervised labeled data to train two classifiers. From the results in Table 5.3 for the PRECOG dataset and as expected, the classifier trained with data manually labeled performs better, using the semi-unsupervised labeled data resulted in a decrease of performance of 1.8%. This loss in performance is acceptable if we take into account that with this approach we can build a framework capable of recognizing and predicting actions in a semi-unsupervised fashion.

TABLE 5.3: Accuracy (%) comparison of the proposed prediction method for the PRECOG and the CAD-120 dataset, trained with ground-truth data vs data labeled with our semi-unsupervised labeling method using conditional random fields.

Dataset	Manually labeled data	Semi-unsupervised labeled data
PRECOG	91.7%	89.9%
CAD-120	86.27%	54.08%

Parallel work from Koppula and Saxena (2016) also addresses activity anticipation, but in a different way. While we attempt to predict the next possible action that might occur, independent of any measure of time, they try to perform activity anticipation for a specific amount of time. They reach activity anticipation accuracies of 75.4%, 69.2% and 58.1% for anticipation times of 1, 3 and 10 seconds, respectively. Although our approach and objective are distinct, we will

evaluate and experiment on their CAD-120 human activity RGB-D dataset. We report the results obtained by 10-fold cross validation by averaging across the folds. Table 5.3 also shows the metrics for anticipating the next possible sub-activity on the CAD-120 dataset. Using ground-truth data for training CRF we managed an average anticipation accuracy of 86.27%. Although our results present a higher anticipation accuracy than the results obtained by Koppula and Saxena (2016) it would be incorrect to say that we outperformed their approach since we propose to achieve different goals.

The confusion matrix for the action prediction results for the CAD-120 dataset using ground-truth labeled data is shown in Table 5.4, where it is possible to observe that even with a greater variation of sub-activities our approach is able to anticipate future possibilities based on the history of actions performed.

TABLE 5.4: Confusion matrix for action prediction performed on the CAD-120 dataset where the models were created with data manually labeled.

reaching	moving	placing	drinking	eating	opening	closing	pouring	
94.24%	1.67%	2.41%	0.00%	1.27%	0.40%	0.00%	0.00%	reaching
9.28%	77.88%	1.17%	0.00%	0.00%	11.67%	0.00%	0.00%	moving
0.00%	0.00%	99.72%	0.00%	0.28%	0.00%	0.00%	0.00%	placing
0.00%	0.00%	18.56%	73.20%	8.25%	0.00%	0.00%	0.00%	drinking
0.00%	0.00%	3.51%	0.00%	96.49%	0.00%	0.00%	0.00%	eating
2.66%	5.56%	2.17%	0.00%	0.00%	89.61%	0.00%	0.00%	opening
6.49%	0.00%	0.00%	0.00%	0.00%	12.99%	80.52%	0.00%	closing
0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	pouring

As for Table 5.5 which illustrates the confusion matrix when using the semi-unsupervised labeled data to train the action recognition classifiers, the difference is obvious and it is immediately noticeable that our approach struggles if the training data is incorrectly labeled. The current state of our semi-unsupervised labeling approach invalidates the usage of semi-unsupervised labeling data to perform activity prediction since the average prediction accuracy obtained was 54.08%. This reveals a limitation of our approach, the success of each module of the application pipeline is directly related to the performance of the previous module. The performance of the activity recognition module depends on the accuracy of the semi-unsupervised labeling method, and the performance of the activity prediction module is affected by the accuracy of the activity recognition module. For this

prediction approach to be successful the training data has to be labeled correctly and the actions have to be correctly recognized in real-time.

TABLE 5.5: Confusion matrix for action prediction performed on the CAD-120 dataset where the models were created with data labeled by action recognition classifiers that were trained with data labeled by our semi-unsupervised labeling method.

reaching	moving	placing	drinking	eating	opening	closing	pouring	
80.22%	7.91%	4.95%	0.63%	2.79%	2.07%	0.18%	1.26%	reaching
12.77%	73.02%	2.58%	0.38%	1.29%	8.81%	0.23%	0.91%	moving
9.60%	11.30%	70.74%	0.77%	2.32%	3.87%	0.15%	1.24%	placing
23.29%	17.08%	19.88%	24.22%	8.07%	3.42%	0.62%	3.42%	drinking
14.80%	15.05%	10.71%	1.79%	52.04%	4.08%	0.26%	1.28%	eating
19.12%	26.89%	8.40%	1.26%	2.73%	39.50%	0.84%	1.26%	opening
25.55%	25.91%	13.14%	3.28%	6.57%	10.95%	12.77%	1.82%	closing
22.87%	23.97%	10.47%	1.65%	4.13%	7.71%	0.55%	28.65%	pouring

5.5 Discussion

In this chapter, we considered the problem of early activity recognition and activity prediction. In Chapter 4 we presented the skeleton visualizer from the PRECOG application where it is possible to observe and classify in real-time the actions being performed by a subject. In this chapter we implemented and added the action prediction module to the application pipeline. The application is now capable of performing action recognition and prediction in real-time (Figure 5.4). In a real-world application the system would be trained and configurable to respond to certain observable events/actions and act accordingly.

Encouraging results were obtained using manually labeled data to train the prediction classifiers. We observed that the prediction accuracy depends on the data, in this case the complexity of the actions, and also, depends highly on the performance of the previous modules of the pipeline. The performance of the prediction module depends on the ability of the recognition module to correctly recognize an action, which in turn depends on the temporal segmentation and labeling accuracy to generate a good training set.

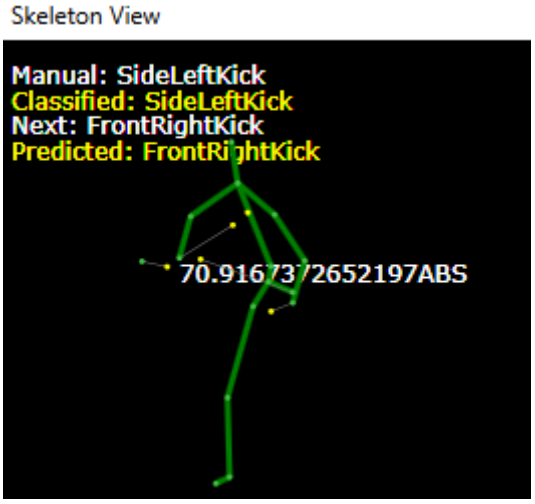

FIGURE 5.4: Screenshot of the PRECOG application performing real-time action recognition and prediction. The labels in white refer to the ground-truth labeled action and prediction. The labels in yellow refer to the recognized and the predicted action.

We consider that we were able to answer our research question which was to create a system capable of performing real-time action recognition and prediction. We demonstrated that it is possible to predict the next possible action based on the current recognized action and the history of the previous recognized actions. The sequence of human sub-activities was modeled using n-grams and conditional random fields in a POS fashion as in NLP problems.

This approach has some limitations that could be addressed in the future work. This kind of prediction will only work if the subject is performing known actions to the recognition classifier and those actions can be used to create a sequence of actions known to the prediction classifier. If an unknown action is performed, how should the system react? This said, it is impossible for a system to try and predict every possible future action, as the possibilities are infinite. Furthermore, the system's ability to recover from mislabeled actions is still poor, although a semi-supervised labeling (as explained in Chapter 4), and possibly retraining after batches of corrections could improve results with minor user intervention. In the event of mislabeling an action, the incorrect action will be added to the history of actions, leading the classifier to make an incorrect prediction. The proposed system could be applied to scenarios where certain behaviors are expected and

can be anticipated (collaborative environments for humans and robots, industry automation, etc).

Chapter 6

Conclusions and Future Work

"Education is the passport to the future, for tomorrow belongs to those who prepare for it today."

Malcolm X

6.1 Conclusions

The ability to analyze images or videos and understand humans and their surroundings has multiple real-world applications such as surveillance systems, self-driving cars, robot assistants in collaborative environments, etc. Human observation and understanding is a natural task that we humans perform with relative ease. On the other hand, replicating the same task with a computer is extremely complex and presents multiple challenges. These challenges include, inadequate environment and recording settings, temporal variations, context, human intention, infinite number of possible actions, ambiguity of actions, occlusions, noise and the cost of labeling large amounts of data required for training. In this dissertation, we addressed some of these challenges by proposing a framework for human activity recognition and prediction using machine learning algorithms.

In our approach, we use the 3D position of the skeleton joints captured by the Kinect sensor to compute features that model human actions and allow us to train classifiers specialized in activity recognition and prediction. The action recognition and prediction process is described by a hierarchical model through four levels of abstraction. A bottom-up approach, ranging from low-level data acquisition from the sensor, feature extraction and feature engineering to design the input of the classification algorithms, low-level or atomic action recognition to high-level action prediction. We recorded and published a new RGB-D dataset, called PRECOG dataset, containing 72 high-level sequences of aggressive actions with male and female subjects. Unlike most of the available datasets which describe isolated actions, the PRECOG dataset is hierarchical and sequential as expected in real-life situations.

Machine learning requires large amounts of labeled data for training. "Big data" is hard to organize, analyze and label. The cost of labeling large amounts of data is one of the main concerns for ML applications. To address this issue, we proposed a method to perform semi-unsupervised labeling of human activity in RGB-D videos. We demonstrated how temporal segmentation, clustering and filtering techniques can be combined to achieve semi-unsupervised labeling of human activity. The advantage of this proposal is that instead of annotating a whole dataset, the human judge only has to label the classes that were identified by our method. We compared the performance of several supervised classifiers used in recent work by other authors to recognize human activity. This work clarified the difference between using manually versus semi-unsupervised labeled data, where the goal was to ascertain the impact of the noise introduced by semi-unsupervised labeling on action classification. Our results showed that, for a dataset of simple combat actions, captured with a standard Kinect sensor with no special acquisition conditions, a temporal segmentation and clustering algorithm can be used to label identical actions performed by different users. Encouraging results were obtained with our PRECOG dataset, as for the CAD-120 dataset the results were less favorable. The CAD-120 dataset contains more ambiguous actions which highlights how challenging it is to create a general method that scales to multi-classes of

actions. A task that even humans have difficulties in performing accurately when dealing with ambiguous actions and have to resort to contextual information to aid the decision.

We presented two distinct approaches for human activity recognition: temporal segment action recognition and frame-by-frame action recognition. For each approach we presented a series of validation experiments: comparison between well-known classifiers, multi-class classifiers versus binary classifiers and measured the impact of using training data labeled in a semi-unsupervised fashion versus data manually labeled by a human judge. In these experiments, the frame-by-frame action recognition approach (employing binary classifiers and a voting policy for the best action based on the classification error with data manually labeled), is able to overcome state-of-the-art results in datasets with similar actions with average precisions of 0.973 and 0.982 for the PRECOG and CAD-120 datasets respectively. This approach has the ability to perform action recognition right from the first frames of the temporal segment. Using data labeled with our semi-unsupervised labeling method brought a decrease of average precision in action recognition of 0.150 for the PRECOG dataset and a decrease of 0.359 for the CAD-120 dataset, highlighting the importance of having data correctly labeled for training. These experiments allowed us to observe and validate the accuracy, robustness and readiness of our approach, all key characteristics required by real-time monitoring systems.

Humans have the ability to effortlessly anticipate a given situation into multiple future possibilities. This is a very challenging task for a computer since the possibilities are endless. While we perform our daily activities, we repeat certain patterns of sequences of actions. Instead of trying to predict each possible future, we proposed a method that uses CRF to recognize patterns of actions to perform activity prediction. Due to the wide application of CRF in labeling sequential data, we saw fit their application to label structured data such as sequences of actions. Unlike other contemporary and parallel approaches, that take into account the context of the scene or perform object recognition in order to obtain more information, our approach relies solely on the features extracted from the

movement of the joints of the subject's skeleton to recognize and predict human activity. We presented several validation experiments and obtained the best performance when using manually labeled data for training and obtained an average prediction accuracy of 91.7% and 86.27% for the PRECOG and CAD-120 datasets respectively. Although we obtained a higher score in performance than the state of the art approach, this score is for predicting the next immediate possible action for every captured frame, while the state of the art approach predicts seconds ahead in the future. Depending on the application, one type of prediction can be more adequate than the other.

In summary, we have demonstrated how 3D sensing technology, feature engineering and machine learning enabled the resolution of a very challenging task which is to recognize and predict human activity. We validated our approach with multiple experiments in tasks similar to those used in the state of the art. In some scenarios we have outperformed the state-of-the-art in human activity understanding and matched some different contemporary approaches. The ability to recognize and anticipate what a person might do next has a myriad of applications which, with the current tendency of big-data generation and retrieval, will only improve with time.

6.2 Future Work

One of the limitations of our approach is that a human judge is always required to assign a label to a set of temporal segments clustered by a clustering method. Completely unsupervised labeling of human activity data is very challenging. It requires accurate general-purpose temporal segmentation and correct clustering of actions (even if they are ambiguous). A generic and scalable method capable of performing unsupervised labeling of human activity, in an accurate way, is what future research should aim for (regardless of the dataset and the classes of actions). One possible way to improve our semi-unsupervised labeling method would be to filter the instances that are closest to the centroid of the cluster and use them as

candidates to train the classifiers. The instances that are above a certain distance from the centroid of the cluster could be labeled off-line by the previously trained classifier.

Another limitation of the approach proposed in this thesis, is the inability of our activity prediction method to handle unknown actions. If an action occurs that the system does not recognize, a domino effect might occur and the system might not be able to cope. A failure detection event must be implemented to allow the system to restart action anticipation as soon as a new activity is classified. In order to perform human activity prediction with our approach, a certain order of sub-activities and patterns is expected. This limits the range of applications where our system can be deployed. We expected that some of the ML algorithms that were used, would be able to overcome the noise introduced by the semi-unsupervised labeling method (since they are known to behave relatively well in noisy scenarios), unfortunately this did not happen.

With the current reform of personal data protection rules in the EU, the collection and management of personal data has to abide to strict legal conditions. In the future and depending on the application, HAR systems will have to ensure the privacy of the subjects being monitored. Data anonymization has to be performed and the sensors used might only extract anonymous features. For example RGB-D sensors manage to preserve much more privacy than traditional video cameras if the RGB feed is ignored. Higher level of privacy can be obtained by using the skeleton joints to represent a person. Although the recent advances in RGB-D sensors fostered the development of promising approaches, an improvement of the skeleton tracking process is required in order to deal with joint occlusions and frame loss which affect the accuracy and capability of response of a HAR system based in RGB-D technology.

The future research of this area will be highly encouraged and dictated by applications. As the need for surveillance of public facilities, development of autonomous vehicles and autonomous robots increases, we believe that human motion analysis and anticipation will become part of our everyday lives.

Bibliography

J. K. Aggarwal and M. S. Ryoo. Human Activity Analysis: A Review. *ACM Computing Surveys*, 43(3):1–43, 2011.

J. F. Allen. Maintaining Knowledge about Temporal Intervals. In *Readings in Qualitative Reasoning About Physical Systems*, pages 361–372. 2013.

F. I. Bashir, A. a. Khokhar, and D. Schonfeld. Object trajectory-based activity classification and recognition using hidden Markov models. *IEEE Transactions on Image Processing : IEEE Signal Processing Society*, 16(7):1912–9, 2007.

D. M. Blei, A. Y. Ng, M. I. Jordan, H. M. Wallach, G. E. Hinton, S. Osindero, and Y.-W. Teh. Conditional random fields: An introduction. *Neural Computation*, 18:1–9, 2004.

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

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

E. Cippitelli, S. Gasparrini, E. Gambi, and S. Spinsante. A human activity recognition system using skeleton data from rgbd sensors. pages 21–. Hindawi Publishing Corporation, 2016.

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

P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proceedings of the 14th International Conference on Computer Communications and Networks*, ICCCN '05, pages 65–72, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7803-9424-0.

M. A. H. Eibe Frank and I. H. Witten. The WEKA Workbench. In *Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*. Morgan Kaufmann, 2016.

M. Elgendi, F. Picon, and N. Magenant-Thalmann. Real-time speed detection of hand gesture using, kinect. In *Proc. Workshop on Autonomous Social Robots and Virtual Humans, The 25th Annual Conference on Computer Animation and Social Agents (CASA 2012)*, pages 1–15, 2012.

E. Fox, E. Sudderth, M. Jordan, and A. Willsky. Nonparametric Bayesian learning of switching linear dynamical systems. *Advances in neural information processing systems*, 21:457–464, 2009.

S. Gaglio, G. L. Re, and M. Morana. Human Activity Recognition Process Using 3-D Posture Data. *IEEE Transactions on Human-Machine Systems*, 45(5):586–597, 2015.

D. Gavrila. The Visual Analysis of Human Movement: A Survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. volume 29, pages 2247–2253, Washington, DC, USA, Dec. 2007. IEEE Computer Society.

M. a. Gowayyed, M. Torki, M. E. Hussein, and M. El-Saban. Histogram of Oriented Displacements (HOD): Describing trajectories of human joints for action recognition. In *International Joint Conference on Artificial Intelligence*, volume 25, pages 1351–1357. AAAI Press, 2013.

A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos.

In *Conference on Computer Vision and Pattern Recognition*, pages 2012–2019. IEEE Press, 2009.

J. A. Hartigan and M. A. Wong. A K-Means Clustering Algorithm. *Applied Statistics*, 28(1):100–108, 1979.

K. P. Hawkins, N. Vo, S. Bansal, and A. F. Bobick. Probabilistic human action prediction and wait-sensitive planning for responsive human-robot collaboration. In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference*, pages 499–506. IEEE, 2013.

M. Hoai and F. De la Torre. Maximum margin temporal clustering. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, pages 1–9, 2012.

M. Hoai and F. De La Torre. Max-margin early event detectors. *International Journal of Computer Vision*, 107(2):191–202, 2014.

M. Hoai, Z.-Z. Lan, and F. D. L. Torre. Joint segmentation and classification of human actions in video. *Computer Vision Pattern Recognition 2011*, pages 3265–3272, 2011.

T. Hofmann and J. M. Buhmann. Competitive learning algorithms for robust vector quantization. *IEEE Transactions on Signal Processing*, 46(6):1665–1675, 1998.

M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *International Joint Conference on Artificial Intelligence*, pages 2466–2472. AAAI Press, 2013.

S. S. Intille and A. F. Bobick. A Framework for Recognizing Multi-agent Action from Visual Evidence. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference*, number 489, pages 518–525. AAAI Press, 1999.

Y. A. Ivanov and A. F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):852–872, 2000.

J. O'Rourke and N. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):522–536, 1980.

D. Jardim, L. Nunes, and M. S. Dias. Human activity recognition and prediction. In *Proceedings of the Doctoral Consortium in The International Conference on Pattern Recognition Applications (ICPRAM)*, page pp. 24–32. SCITEPRESS Digital Library, 2015.

D. Jardim, L. Nunes, and M. Dias. Human activity recognition from automatically labeled data in rgb-d videos. In *Computer Science and Electronic Engineering (CEEC), 2016 8th*, pages 89–94. IEEE, 2016a.

D. Jardim, L. Nunes, and M. S. Dias. Impact of automated action labeling in classification of human actions in rgb-d videos. In *22nd European Conference in Artificial Intelligence: ECAI 2016*, page in press. IOS Press, 2016b.

D. Jardim, L. Nunes, and M. S. Dias. Automatic human activity segmentation and labeling in rgb-d videos. In *Intelligent Decision Technologies: KES-IDT 2016*, pages SIST 56, p. 383. Springer International Publishing, 2016c.

D. Jardim, L. Nunes, and M. Dias. Predicting human activities in sequences of actions in rgb-d videos. In *Ninth International Conference on Machine Vision (ICMV 2016)*, pages 103410C–103410C–5. SPIE Digital Library Proceedings, 2017.

C. Jia, Y. Kong, Z. Ding, and Y. R. Fu. Latent tensor transfer learning for rgb-d action recognition. In *Proceedings of the ACM International Conference on Multimedia*, pages 87–96. ACM Press, 2014.

D. Jurafsky and J. H. Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. *Speech and Language Processing An Introduction to Natural Language Processing Computational Linguistics and Speech Recognition*, 21:0–934, 2009.

R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83(1):95–108, 1961.

M. Kaur and U. Kaur. Comparison Between K-Mean and Hierarchical Algorithm Using Query Redirection. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(7):1454–1459, 2013.

C. G. Keller, T. Dang, H. Fritz, A. Joos, C. Rabe, and D. M. Gavrila. Active pedestrian safety by automatic braking and evasive steering. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1292–1304, 2011.

Kinect for Windows SDK 1.8. Skeletal tracking, 2017. URL `https://msdn.microsoft.com/en-us/library/hh973073.aspx`. [Online; accessed March 6, 2017].

K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. Activity forecasting. In *European Conference on Computer Vision*, pages 201–214. Springer, 2012.

Y. Kong and Y. Fu. *Human Activity Recognition and Prediction*. 2015. ISBN 9783319270043.

H. S. Koppula and A. Saxena. Anticipating Human Activities Using Object Affordances for Reactive Robotic Response. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):14–29, 2016.

H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from RGB-D videos. *The International Journal of Robotics Research*, 32:951–970, 2013.

M. Kubat. Neural networks: a comprehensive foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7. In *The Knowledge Engineering Review*, volume 13, pages 409–412. Cambridge Univ Press, 1999.

L. A. Leiva and E. Vidal. Warped k-means: An algorithm to cluster sequentially-distributed data. volume 237, pages 196–210. Elsevier, 2013.

K. Li, J. Hu, and Y. Fu. *Modeling Complex Temporal Composition of Actionlets for Activity Prediction*, pages 286–299. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3D points. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, pages 9–14, 2010.

J. Mainprice and D. Berenson. Human-robot collaborative manipulation planning using early prediction of human motion. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 299–306, 2013.

Microsoft Corporation. Kinect coordinate spaces, 2016a. URL `https://msdn.microsoft.com/en-us/library/hh973078.aspx`. [Online; accessed January 17, 2016].

Microsoft Corporation. Kinect for windows sensor components and specifications, 2016b. URL `https://msdn.microsoft.com/en-us/library/jj131033.aspx`. [Online; accessed June 21, 2016].

Microsoft Corporation. Skeleton position and tracking state, 2016c. URL `https://msdn.microsoft.com/en-us/library/jj131025.aspx`. [Online; accessed June 21, 2016].

Microsoft Corporation. Kinect for windows sensor v2 components and specifications, 2017. URL `https://developer.microsoft.com/en-us/windows/kinect/develop`. [Online; accessed January 17, 2017].

D. Moore and I. Essa. Recognizing multitasked activities from video using stochastic context-free grammar. *Innovative Applications of Artificial Intelligence Conferences*, pages 770–776, 2002.

R. Nevatia, T. Zhao, and S. Hongeng. Hierarchical Language-based Representation of Events in Video Streams. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, volume 4. IEEE, 2003.

N. T. Nguyen, D. Q. Phung, S. Venkatesh, and H. Bui. Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 955–960. IEEE, 2005.

S. Nikolaidis, P. Lasota, G. Rossano, C. Martinez, T. Fuhlbrigge, and J. Shah. Human-robot collaboration in manufacturing: Quantitative evaluation of predictable, convergent joint action. In *44th International Symposium on Robotics, (ISR)*, 2013.

S. Nirjon, C. Greenwood, C. Torres, S. Zhou, J. a. Stankovic, H. J. Yoon, H. K. Ra, C. Basaran, T. Park, and S. H. Son. Kintense: A robust, accurate, real-time and evolving system for detecting aggressive actions from streaming 3D skeleton data. In *International Conference on Pervasive Computing and Communications*, pages 2–10. IEEE Press, 2014.

W. Niu, J. Long, D. Han, and Y. F. Wang. Human activity detection and recognition for video surveillance. In *International Conference on Multimedia and Expo*, pages 719–722. IEEE Press, 2004.

N. Oliver, E. Horvitz, and A. Garg. Layered Representations for Human Activity Recognition. *ICMI '02 Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, pages 1–6, 2002.

L. Onofri, P. Soda, M. Pechenizkiy, and G. Iannello. A survey on using domain and contextual knowledge for human activity recognition in video streams, 2016. ISSN 09574174.

K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77, 2007.

A. Pentland and A. Liu. Modeling and Prediction of Human Behavior. *Neural Computation*, 11(1):229–242, 1999.

C. S. Pinhanez and A. F. Bobick. Human action detection using pnf propagation of temporal constraints. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 898–904. IEEE Press, 1998.

J. C. Platt. Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In *Advances in Kernel Methods - Support Vector Learning*, pages 185 – 208. MIT Press, 1998.

M. Popa, A. Kemal Koc, L. J. M. Rothkrantz, C. Shan, and P. Wiggers. Kinect sensing of shopping related actions. In *Communications in Computer and Information Science*, volume 277 CCIS, pages 91–100. Springer, 2012.

L. R. Rabiner and B. H. Juang. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, 3(1):4–16, 1986. ISSN 07407467.

K. Ramirez-Amaro, M. Beetz, and G. Cheng. Transferring skills to humanoid robots by extracting semantic representations from observations of human activities. *Artificial Intelligence*, 2015.

R. F. Rashid. Towards a system for the interpretation of moving light displays. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 574–581. IEEE Press, 1980.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning Internal Representations by Error Propagation, 1986.

M. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. *International Conference on Computer Vision*, pages 1036–1043, 2011.

M. S. Ryoo and J. K. Aggarwal. Recognition of composite human activities through context-free grammar based representation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1709–1716. IEEE, 2006.

M. S. Ryoo and J. K. Aggarwal. Semantic Representation and Recognition of Continued and Recursive Human Activities. In *International Journal of Computer Vision*, volume 82, pages 1–24. Springer International Publishing, 2009.

A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1010–1019. IEEE, 2016.

Q. Shi, L. Cheng, L. Wang, and A. Smola. Human action segmentation and recognition using discriminative semi-Markov models. *International Journal of Computer Vision*, 93(1):22–32, 2011.

T. Starner, J. Weaver, and A. Pentland. Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video. In *Transactions on Pattern Analysis and Machine Intelligence*, volume 20, pages 1371–1375. IEEE Press, 1998.

J. Sung, C. Ponce, B. Selman, and A. Saxena. Human Activity Detection from RGBD Images. *IEEE International Conference on Robotics and Automation*, pages 842–849, 2011.

C. Sutton and A. McCallum. An Introduction to Conditional Random Fields. *Machine Learning*, 4(4):267–373, 2010.

TAV IT. Tav it airport information management, 2017. URL `https://www.copybook.com/companies/tav-it-airport-information-management`. [Online; accessed March 9, 2017].

Tonye Ogele CNX. Flexion and extension, 2017. URL `https://commons.wikimedia.org/wiki/File:Body_Movements_I.jpg`. [Online; accessed March 1, 2017].

S. Vishwakarma and A. Agrawal. A survey on activity recognition and behavior understanding in video surveillance. In *Visual Computer*, volume 29, pages 983–1009, 2013. ISBN 0037101207526. doi: 10.1007/s00371-012-0752-6.

M. Vrigkas, C. Nikou, and I. A. Kakadiaris. A review of human activity recognition methods. *Frontiers in Robotics and AI*, 2:28, 2015.

V. T. Vu, F. Bremond, and M. Thonnat. Automatic video interpretation: A novel algorithm for temporal scenario recognition. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 1295–1300, 2003.

M. Wachter and T. Asfour. Hierarchical segmentation of manipulation actions based on object relations and motion characteristics. *Proceedings of the 17th International Conference on Advanced Robotics, ICAR 2015*, 270273:549–556, 2015.

J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1290–1297, 2012.

Waymo Alphabet. Waymo, 2017. URL `https://waymo.com/`. [Online; accessed March 9, 2017].

C. Wolf, J. Mille, E. Lombardi, O. Celiktutan, M. Jiu, E. Dogan, G. Eren, M. Baccouche, E. Dellandrea, C.-E. Bichot, C. Garcia, and B. Sankur. Evaluation of video activity localizations integrating quality and quantity measurements. In *Computer Vision and Image Understanding*, volume 127, pages 14–30. Elsevier, 2014.

J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden Markov model. In *Computer Vision and Pattern Recognition*, pages 379–385. IEEE Press, 1992.

E. Yu and J. K. Aggarwal. Detection of Fence Climbing from Monocular Video. In *18th International Conference on Pattern Recognition*, volume 1, pages 375–378. IEEE Press, 2006.

F. Zhou, F. D. L. Torre, and J. Hodgins. Hierarchical Aligned Cluster Analysis (HACA) for Temporal Segmentation of Human Motion. In *IEEE Transactions*

on Pattern Analysis and Machine Intelligence, volume 35, pages 1–40. Citeseer, 2013.

B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *International Conference on Intelligent Robots and Systems, (IROS)*, pages 3931–3936. IEEE, 2009.