Geometric + Kinematic Priors and Part-based Graph Convolutional Network for Skeleton-based Human Action Recognition

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering
(by Research)

by

Kalpit Thakkar
201201071
kalpit.thakkar@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
March 2019

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “Geometric + Kinematic Priors and Part-based Graph Convolutional Network for Skeleton-based Human Action Recognition” by Kalpit Thakkar, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. Narayanan, P J

To

My relentless companion, Love.

&

My greatest gift, Life.

PREFACE

When I was born, tears of joy were rolling down the cheeks of my mother and my father, seeing their brave first child making an entry into this world. To make my journey and understanding about life extraordinary, they made me a spiritual boy. They taught me music, football, art and gifted me a mindset like none other, while always preaching that I stay down to earth. I recieved education in the best school of the city, developed strong and heartwarming friendships, learned sportmanship at its finest and as a result, I decided at the age of sixteen that I wanted to inspire and educate. My life has since been a life driven by the dream of becoming a professor.

Joining an IIT-JEE coaching class in XI grade marked the inception of my journey of seeking intellectual excellence required to achieve my dream. They told me there is no substitute to hard work, but I did not understand that they meant "consistent" hard work. Clearly, I encountered a hiccup in my journey, but who does not? I did not do very well on my IIT-JEE and the chance of going to an IIT had gone up in flames. But I was not done yet. I did really well on AIEEE and got the next best thing – IIIT Hyderabad, CSE. Then, I repeated the mistake of putting in hard work but forgetting the adjective "consistent" which really counts; for the first year. What happened next felt like a miracle: I became sincere. I became a "consistent" hard worker. I fell in love. I went crazy in love. Most importantly, I stayed a footballer and made sure it kept me grounded. However, after four years of Bachelors, I felt that I did not make the most of the intellectual & emotional roller coaster that college life is and I wanted to learn more. Also, research was necessary for achieving my dream of becoming a professor. To top it all, I loved freedom of deciding my own tasks and goals, unlike the dynamics of a corporate culture. I made a lot of mistakes too, and I saw research as a way of redeeming myself, of being on an intellectual journey that really mattered: understanding yourself and the world around you.

In accord to the revelations mentioned above, I converted to a Dual Degree program at IIIT Hyderabad, in my last semester. And hence began the journey towards bringing this magnificent manuscript into existence, which is a testament to my quest for knowledge and redemption. And just like I said before, I am a spiritual boy – the journey matters more to me than the final outcome. This manuscript is a documentation of my amazing journey, which entails vision and learning that goes deep. The most exciting part is that this journey has not yet ended and probably never will. And, I was never alone. My journey has been affected by numerous personalities. And I would like to take a moment to thank them.

My mother, Nimisha, has been the warrior who made me who I am today. She worked tirelessly round the clock for me: dropping and picking me up from school, football practices, music classes and art classes. She is my hero. My father, Chetan, has been a role model who always kept me grounded and did what was necessary. He pushed me to do good and taught me to not get comfortable being mediocre, striving for a better tomorrow everday. He is my rock. My younger brother, Kathan, taught me that nothing is impossible if you have the heart to keep going. He has achieved things that surprise me till date and inspire me. He is my inspiration.

My girlfriend, Somya, taught me the wonders of empathy, honesty and introspection. She was by my side during the tumultous college days and kept me going. She is my Wonder Woman. My grandmother, Ramila, always called me once in a while to check up on how I was doing and did not ask for much from my side. Always loved me. My grandfather, Girdharlal, showed me the wonders of selflessness and perseverance. He walked everyday for 10 kms even at the age of 70. Thank you, my family.

I would like to whole heartedly thank my adviser Professor P J Narayanan. He took me under him as a fledgling, asked me the right questions and let me find my answers, making me capable of creating my own path. Also, thanks to Ishit, a fellow Dual Degree student under Prof PJN, who has been a great friend and reviewer. His laugh is really contagious as well as ridiculous. PhD students, Parikshit and Saurabh, also helped me a lot with discussions and I cannot thank them enough. Parikshit was one of the strongest driving forces in a number of decisions I made during my Masters tenure.

My fellow brothers in the "bikerchokras" brotherhood, Nihit and Mohit, thanks for making my birthdays worth remembering as I never got any other chances to celebrate it in college. Thanks for the evergreen travel enthusiasm. "Jati rehje" is going to uncannily stay with me for the rest of my life. My college group of friends: Mohit, Princu, Jigar, Devansh, Ankit and Anubhav, thank you guys for making my college life memorable and mischievous. Nobody uses my stuff now except me, forget about permission. When I was out in the gym working out, my friends Jaspreet and Akash always had my back. Thank you guys. Also, a big thanks to Shikher and Aniruddh for the wonderful soccer memories.

This journey of research has come to a wonderful state of affairs, where I am setting a milestone as well as moving on and embarking on further endeavours for achieving my dream. Everyone who was a part of this journey has affected me in one way or another, and it is all significant. Each and every miniscule decision was significant. I am having a sense of overwhelming gratitude towards all, even if they shared a small moment with me, because it mattered. In the large scheme of things, it all mattered.

Lastly, thanks from the bottom of my heart to my spiritual mentors during my childhood: Jignesh and Rajan. The spiritually involving discussions with you made me rethink my understandings and brought about great revelations. Words are not enough to thank you for making me start playing football, Rajan uncle. That was the best thing that ever happened to me. Thanks to this beautiful game: it taught me lessons of life like no other experience has ever. The feeling of sportsmanship spirit running through you when you play, it made me humble and determined. The feeling of transcendence while playing the game, it kept me down to earth and induced empathy. This game changed the way I looked at my life, myself as well as others.

Thank you very much.
Por vida.

– Kalpit Thakkar

“Live as if you were to die tomorrow. Learn as if you were to live forever.”

– Mahatma Gandhi

Abstract

Videos engulf the media archive in every capacity, generating a rich and vast source of information from which machines can learn how to understand such data and predict useful attributes that can help technologies make human lives better. One of the most significant and instrumental part of a computer vision system is the comprehension of human actions from visual sequences – a paragon of machine intelligence that can be achieved through computer vision. The problem of human action recognition has high importance as it facilitates several applications built around recognition of human actions. Understanding actions from monocular sequences has been studied immensely, while comprehending human actions from a skeleton video has developed in recent times. We propose action recognition frameworks that use the skeletal data (viz. 3D locations of some joints) of human body to learn spatio-temporal representations necessary for recognizing actions.

Information about human actions is composed of two integral dimensions, space and time, along which the variations occur. Spatial representation of a human action aims at encapsulating the configuration of human body essential to the action, while the temporal representation aims at capturing the evolution of such configurations across time. To this end, we propose to use geometric relations between human skeleton joints to discern the body pose relative to the action and physically inspired kinematic quantities in order to understand the temporal evolution of body pose. Spatio-temporal understanding of human actions is thus conceived as a comprehension of geometric and kinematic information with the help of machine learning frameworks. Using a representation inculcating an amalgamation of geometric and kinematic features, we recognize human actions from skeleton videos (S-videos) using such frameworks.

We first present a non-parametric approach for temporal sub-segmentation of trimmed action videos using angular momentum trajectory of the skeletal pose sequence. Meaningful summarization of the pose sequence is a product of the sub-segmentation achieved through systematic sampling of the segments. Descriptors capturing geometric and kinematic statistics encoded as histograms and spread across a periodic range of orientations are computed to represent the summarized pose sequences, which are fed to a kernelized classifier for recognizing actions. This framework for understanding human actions instils the effects of using geometric and kinematic properties of human pose sequences, important to spatio-temporal modeling of the actions. However, a downside of this framework is the inability to scale with availability of large amount of visual data.

To mitigate the impending drawback, we next present geometric deep learning frameworks and specifically, graph convolutional networks, for the same task. Representation of human skeleton as a sparse spatial graph is intuitive and a structured form which lies on graph manifolds. A human action video hence results in the formation of a spatio-temporal graph and graph convolutions facilitate the learning of a spatio-temporal descriptor for the action. Inspired by the success of Deformable Part-based Models (DPMs) for the task of object understanding from images, we propose a part-based graph convolutional network (PB-GCN) that operates on a human skeletal graph divided into parts. Incorporating the culmination of understandings from the success of geometric and kinematic features, we propose to use relative coordinates and temporal displacements of the 3D joints coordinates as features at the vertices in the skeletal graph. Owing to these signals, the prowess of graph convolutional networks is further boosted to attain state-of-the-art performance among several action recognition systems using skeleton videos. In this thesis, we meticulously examine the growth of the idea about using geometry and kinematics, transition to geometric deep learning frameworks and design a PB-GCN with geometric + kinematic signals at the vertices, for the task of human action recognition using skeleton videos.

Contents

List of Figures

Figure Page

List of Tables

Chapter 1

Introduction

1.1 Action Recognition: The Beginning

Since the inception of the pursuit of knowledge about humans through research methodologies, various disciplines including psychology, biology and computer science have invested faculties in the understanding of motions and actions. Analyzing motions and actions has a long history which can be traced back to 500 BC with the dichotomy paradox of the famous Greek philosopher Zeno [15]. A lot of fascination about human motion analysis has developed since. One of the earliest investigations into the nature of human motion was conducted by the contemporary photographers E. J. Marey [16] and E. Muybridge [17] in the 1850s who photographed moving subjects and revealed several interesting and artistic aspects involved in human and animal locomotion [18, 19]. One such pieces of art is shown in Figure 1.1, that shows an exquisite sequence of movements that a man performs while pole vaulting.

It all started with trying to capture pictures of galloping horses to see if all the four hooves come off the ground at any point of time and as it turned out, it does [20] (illustrated in Figure 1.2). After thorough analysis, Muybridge developed a multiple-camera system that could capture human motion and he published the most exquisite sequences of human bodies moving – doing ordinary things: rising, sitting, walking, sweeping [21]. He was the catalyst who brought the inventions of Marey to life and those endeavours inspired the modern motion pictures. On the other hand, in the budding field of neuroscience, the classic moving light display (MLD) experiment of Johansson [22] provided a great impetus to the study and analysis of human motion perception. MLDs consist of bright spots attached to an actor dressed in black, and moving in front of a dark background. The collection of spots carry only 2D and no structural information, since they are not connected. A set of static spots are meaningless to an observer, but the point to be made was that their relative movement created a vivid impression of a person walking, running, dancing, etc. Several people used his MLDs to study and understand how the visual cortex in humans perceives motion cues and recognizes what type of motion (or action) is being performed [23, 24, 25]. These initial efforts paved the way for mathematical modeling of human action

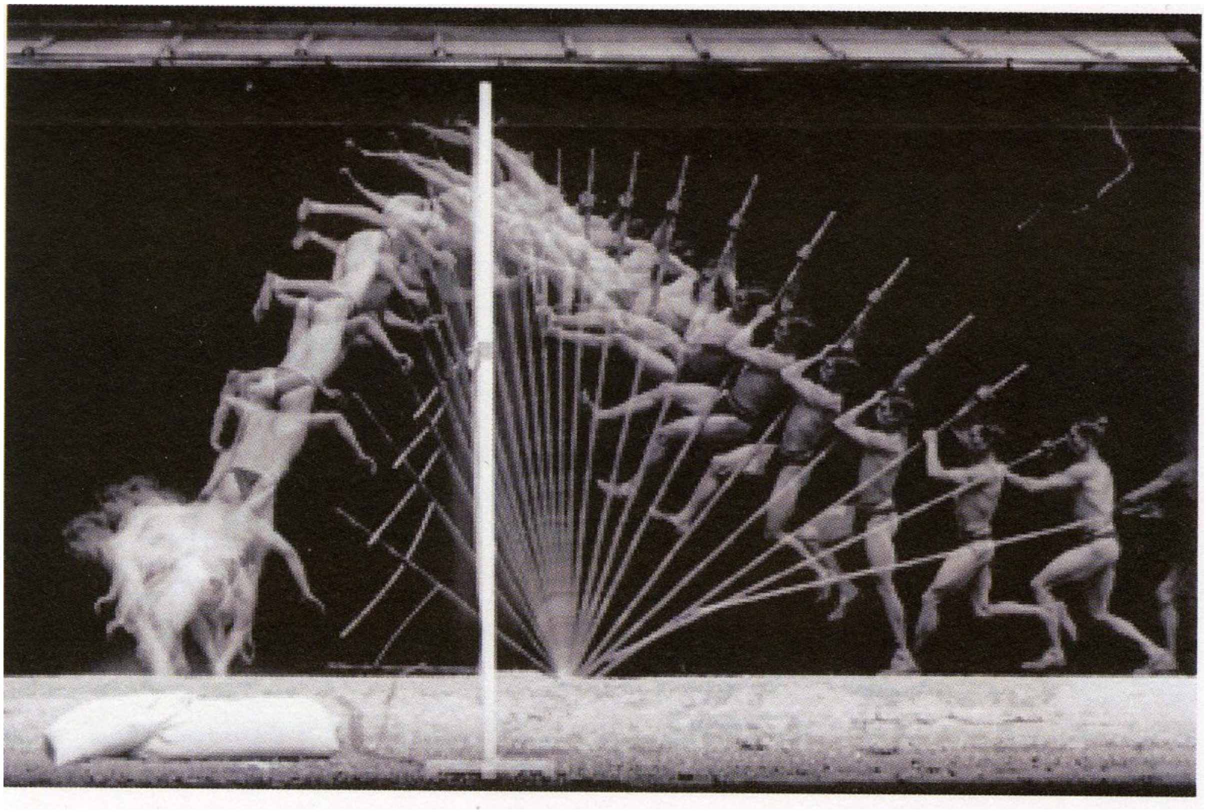


Figure 1.1: Motion sequence captured by Marey that shows the motion of a man performing a pole vault.[1]

[1] Source URL

and automatic recognition, which naturally fall into the realm of computer vision as well as pattern recognition.

1.2 Action Recognition: The Essence

Understanding human behaviour has been an important area of research since the dawn of artificial intelligence [26, 27]. It is a daunting task as it involves learning human emotions and their subtle meanings depending on the context, understanding human actions while they interact with different agents in the environment and learning human motives based on the circumstance, emotion and actions. In order to create a successful machine that can identify and explain human behaviour, all the sub-problems have to be addressed. Each sub-problem has its own challenges and several sub-sub-problems branching out from it. Among several research areas in artificial intelligence and computer vision focusing on humans, viz. detecting humans, tracking humans in videos, pose estimation, human 3D reconstruction, action recognition (or motion recognition), understanding human actions for motion recognition is very important due to its potential far reaching applications such as video surveillance [28, 29], assisted living [27], human computer interaction [30], human-robot interaction [31], self-driving cars [32, 33], etc.

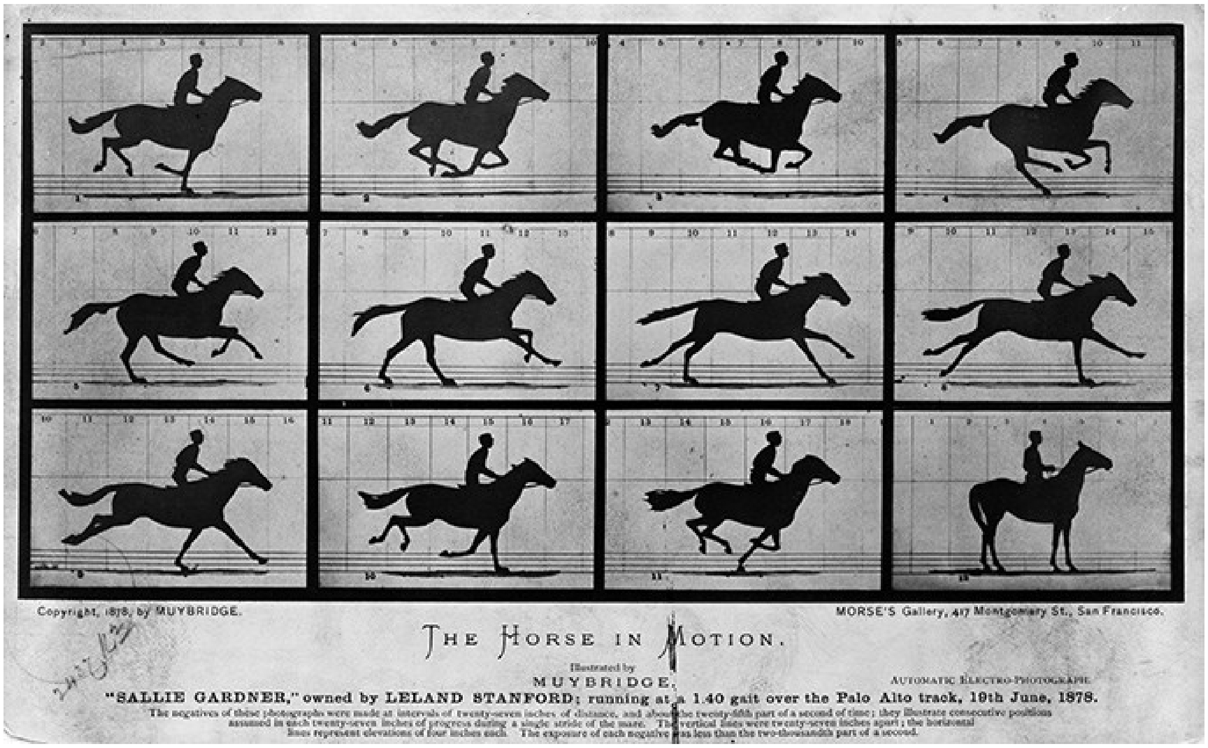

Copyright, 1878, by MUYBRIDGE.
MORSE'S Gallery, 417 Montgomery St., San Francisco.
THE HORSE IN MOTION.
Illustrated by
MUYBRIDGE.
AUTOMATIC ELECTRO-PHOTOGRAPH.
"SALLIE GARDNER," owned by LELAND STANFORD; running at a 1.40 gait over the Palo Alto track, 19th June, 1878.
The negatives of these photographs were made at intervals of twenty-seven inches of distance, and about the twenty-fifth part of a second of time; they illustrate consecutive positions assumed in each twenty-seven inches of progress during a single stride of the mare. The vertical lines were twenty-seven inches apart; the horizontal lines represent elevations of four inches each. The exposure of each negative was less than the two-thousandth part of a second.

Figure 1.2: The sequence of pictures showing a galloping horse, captured by Muybridge. As one can see, all the four hooves come off the ground in one of the pictures.[2]

[2] Source URL

In order to understand the problem of action recognition, let us first understand what does "action" mean: Movements performed by a human range from the single limb movement to joint movements of a group of limbs and body. For instance, kicking a ball is a simple leg movement but jumping to head the ball involves multiple body parts that take part in the movement. Definitions of *action* based on some previous literature include:

- Moeslund and Granum [34], Poppe [35] define action primitives as "an atomic movement that can be described at the limb level". Accordingly, the term action defines a diverse range of movements, from "simple and primitive ones" to "cyclic body movements". The term activity is used to define "a number of subsequent actions", representing a complex movement. For instance, left leg forward is an action primitive of running. Jumping hurdles is an activity performed with the actions starting, running and jumping,

- Turaga *et al.* [36] define action as "simple motion patterns usually executed by a single person and typically lasting for a very short duration (approximately tens of seconds)." Their activity refers to "a complex sequence of actions performed by several humans who could be interacting with each other in a constrained manner." For example, actions are walking or swimming, activities are two persons shaking hands or a football team scoring a goal,

- Chaaraoui *et al.* [27] suggests a that in the context of human behavior analysis, human actions can be broken down into hierarchical elements. The breakdown is based on the level of semantics and the temporal granularity, and considers the "action" in a level between the "motion" and the "activity". Actions are defined as primitive movements (*e.g.*, sitting, walking) that can last up to several minutes and,

- Wang *et al.* [37] suggest that the true meaning of an action lies in "the change or transformation an action brings to the environment", *e.g.*, kicking a ball.

According to the *philosophy of action* from Wikipedia:

> "It is defined as an intentional, purposive, conscious and subjectively meaningful activity."

However, there are certain involuntary human responses caused without their intent as well, which are considered as actions in literature for action recognition research. For example, coughing, sneezing, etc. In the Oxford Dictionary, *action* is defined as "the fact or process of doing something, typically to achieve an aim" and *activity* is "a thing that a person or group does or has done" [38]. For the purposes of explaining the work in this thesis, our definition of action is similar to the one described in [34, 35], which considers actions as a "combination of the articulated motion of several body parts".

Putting all these definitions of *action* together, we can perform a high level categorization of human motions. The complexity and duration of motion involved can be used as basis for broad categorization into four kinds namely gesture, action, interaction and group activity. The class of motions having shortest duration and consist of the basic movement of hand, arm, body or head that emphasizes an idea, emotion, etc. is called a *gesture*. "Hand signs" and "nodding" are some typical examples of gestures. Figure 1.3 shows the gestures made by using hands in order to communicate. An *action* involves the motion of multiple body parts, unlike gestures which involve the motion of few body parts or limbs. When multiple actions are performed one after another, the motion sequence is called an *activity*. When there are two actors involved in the *action*, the other can be either a human or an object. Such a motion sequence is called an *interaction*. Hence, interactions cover two types of actions, one involving human-human interaction and the other involving human-object interaction. "Kicking the other person" and "throwing the ball" are examples of these two types of interactions, respectively. Different activities are shown in Figure 1.4 where each activity involves several sub-actions. *Group activity* is a combination of the various types of motions described above, consisting of more than two human actors and multiple object interactions. "Courtroom trial" and "football practice" are examples of group activities.

1.3 Action Recognition: The Problem

We now define the problem of action recognition. To state the problem in simple terms, given a sequence of images with one or more persons performing an activity, can a system be designed that can automatically recognize what activity is being or was performed? To explain the problem in a

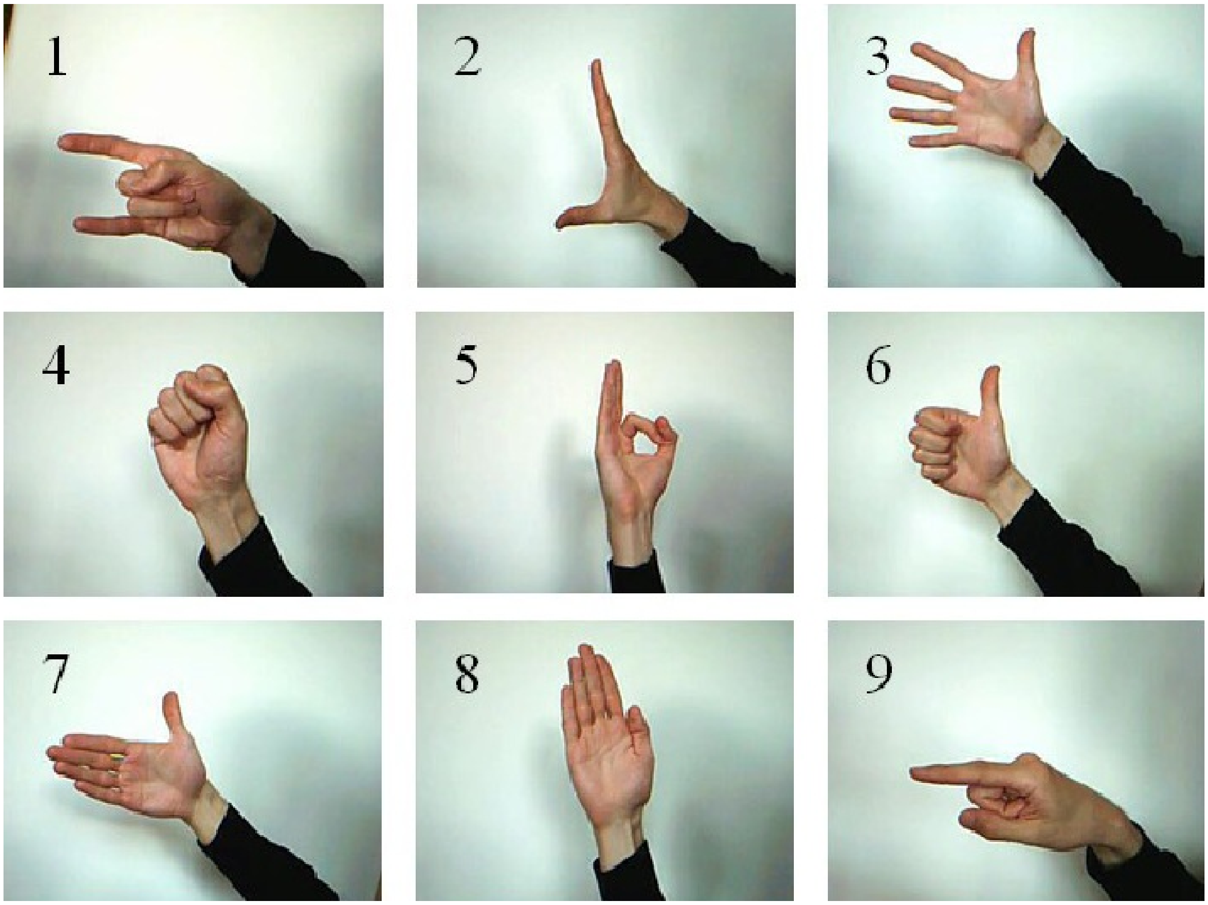
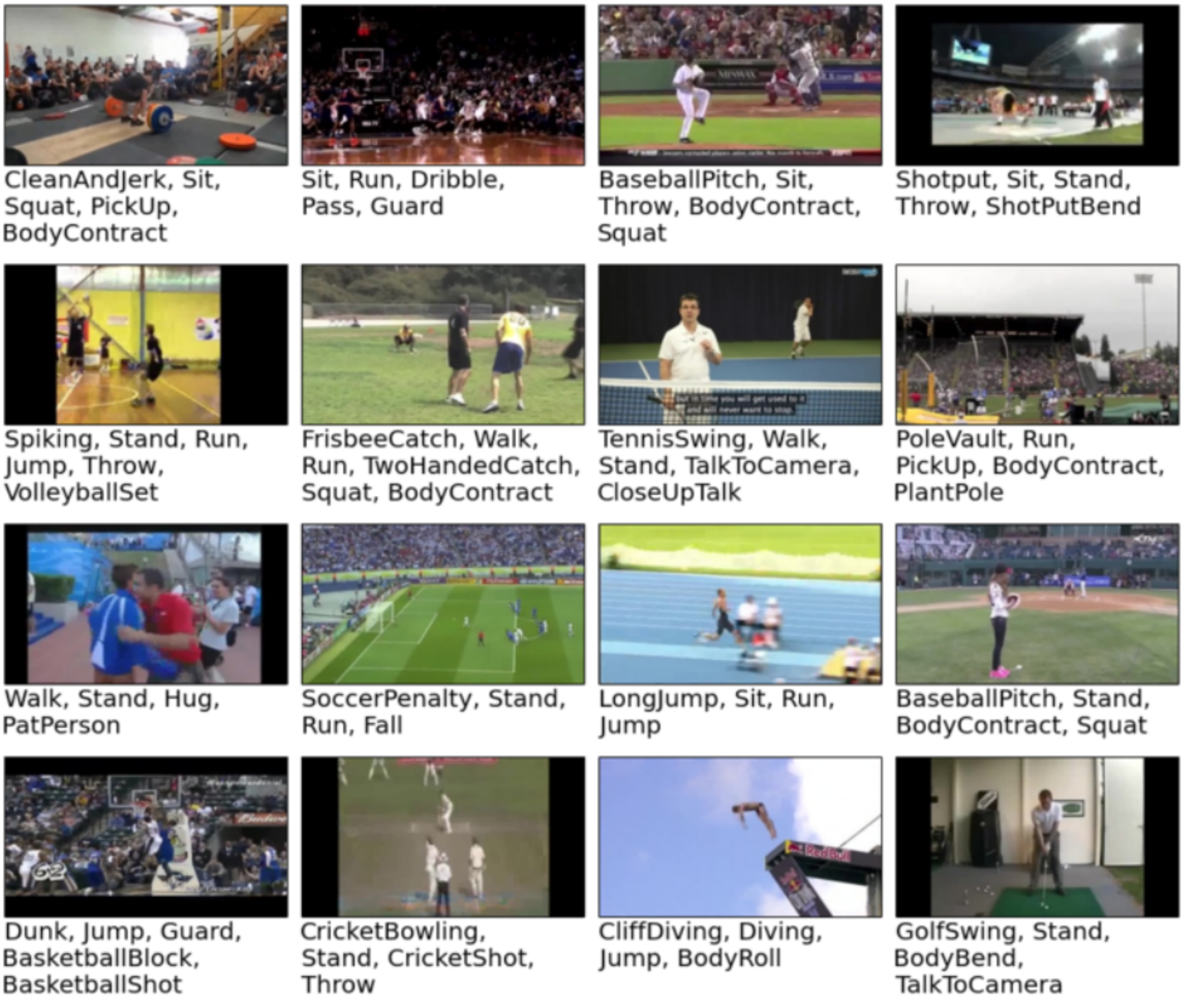


Figure 1.3: Hand sign gestures. Gestures are the basic movements where only a single body part moves.[3]

technical manner: Given some predefined set of actions that are known to the system, can the system be classify an unseen sequence of images containing human motion into one of the actions in the set? On a different line of thought, efforts have been made to explain how motion recognition occurs, based on the seminal work by Johansson on MLDs [22] which perform analysis of human motions using lights as body markers, with two theories [24]: "In the first theory, the visual system performs shape-from-motion reconstruction of the object and then use that to recognize the action. In the second theory, the visual system utilizes motion information directly without performing reconstruction." Hence, the aim of an action recognition system is to identify new, unseen actions based on the set of actions known to it by either reconstruction of human using the motion cues or using the motion cues directly to perform recognition.

The task of understanding human actions has been carried out using various types of input information. Human actions have been recognized from RGB videos [38], which are videos we see in our day to day lives, say, on YouTube. Human actions are also recognized from RGBD videos which contain a depth video in addition to the RGB video. It is typically captured using a sensor-based system like

[3] Source URL



CleanAndJerk, Sit, Squat, PickUp, BodyContract

Sit, Run, Dribble, Pass, Guard

BaseballPitch, Sit, Throw, BodyContract, Squat

Shotput, Sit, Stand, Throw, ShotPutBend

Spiking, Stand, Run, Jump, Throw, VolleyballSet

FrisbeeCatch, Walk, Run, TwoHandedCatch, Squat, BodyContract

TennisSwing, Walk, Stand, TalkToCamera, CloseUpTalk

PoleVault, Run, PickUp, BodyContract, PlantPole

Walk, Stand, Hug, PatPerson

SoccerPenalty, Stand, Run, Fall

LongJump, Sit, Run, Jump

BaseballPitch, Stand, BodyContract, Squat

Dunk, Jump, Guard, BasketballBlock, BasketballShot

CricketBowling, Stand, CricketShot, Throw

CliffDiving, Diving, Jump, BodyRoll

GolfSwing, Stand, BodyBend, TalkToCamera

Figure 1.4: Different types of activities involving interactions with different objects and sub-actions. The name of the sub-actions are marked with the action image still.[4]

[4] Source URL

Kinect, which is different from a RGB camera. There has been research work done in recognizing actions from skeletal data [39], where the 3D locations of human skeleton joints are provided for each frame in the video and architectures are designed to recognize actions using only these skeleton videos. There are several challenges that we need to consider when designing an action recognition system:

- As a human action typically takes place over a period of time and involves multiple body parts, it is important to understand the spatial as well as temporal properties of the action. The spatial properties of an action can be the features of the body represented by a single time instant, while the temporal properties can be understood by tracking the changes in these spatial features across time. Hence, there is a challenge of extracting useful spatio-temporal features which can represent the action information accurately and perform precise action recognition,

- The actions can vary in their time resolutions, meaning that some actions can last for a longer period of time compared to other actions. Some humans might perform the same action slower compared to others and hence the motion information would vary. To handle this, we need to design a system that is invariant to the time resolution changes in same action classes,

- Combining spatial and temporal information about the action sequence is not very straightforward. The spatial features representing the appearance or static human body pose should be effectively combined with the temporal features representing the motion or changes in human body pose. Hence, there is a challenge that a good combination should be learned by the system,

- The action representations learned by the system should be interpretable so that the workings of the system can be well understood. A major challenge in deep learning methods is the interpretability of the ways a system works, without which understanding grows weaker regardless of accuracies of recognition getting stronger.

An action recognition system should address all these challenges and strive to perform well in all the areas. Our first framework deals with all these challenges well but falls back due to the shortcoming of scalability with large training data. Our second framework mitigates this drawback and fulfills the requirements laid out by the challenges mentioned above, in the best manner possible, achieving state-of-the-art performance for action recognition.

In this work, we consider the problem of recognizing actions from skeleton video sequences (S-videos) which provide the 3D locations of n (n varies depending on the device used) joints in the human body tracked using sensors like Kinect or motion capture devices. The information provided by 3D joints locations is very different from RGB and Depth, which is why traditional computer vision or deep learning methods using images as input do not work on them as is. Inputs from RGB and S-videos are illustrated in Figure 1.5, which shows the same action captured from multiple views and four consecutive frames of the same action. Pioneering works in the early action recognition research literature demonstrated the importance of geometric and kinematic features, while using simple classifiers to classify the descriptors representing the action sequences. With the development of deep learning methods including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), the action recognition community started utilizing the potential of these architectures to improve the contemporary state-of-the-art action recognition systems. However, the importance of geometric and kinematic features kept coming back to make a point that these representations are useful as well.

1.4 Action Recognition: Our Framework

Our framework for recognizing actions from S-videos is built around the idea of utilizing geometric and kinematic information of the skeleton motion as a rock-hard foundation of the computation of action representations. In addition, it relies on the fact that the representation of human body which enables our model to learn powerful representations is based on a graph-structured design of human skeleton.

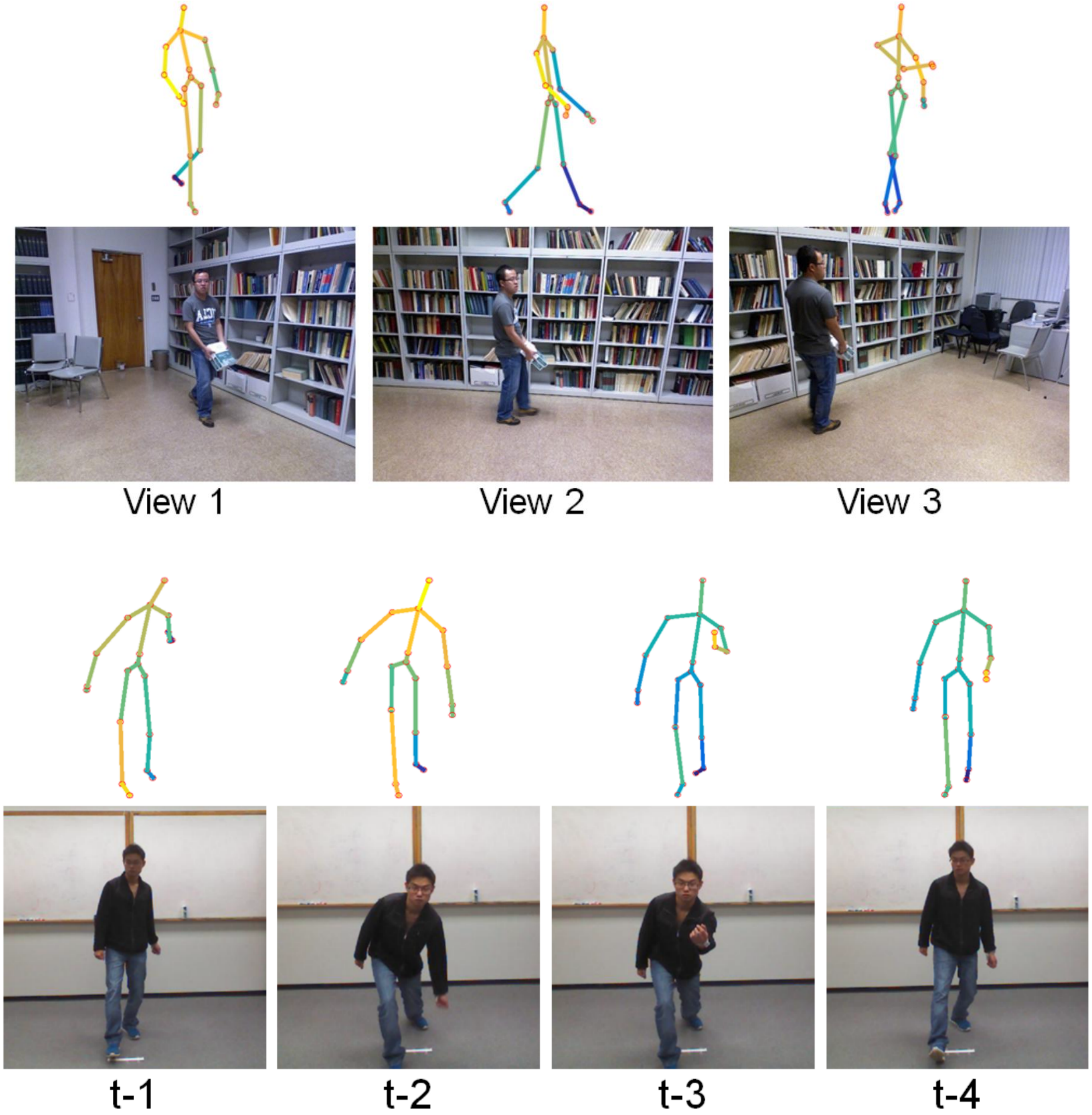


Figure 1.5: **Top**: Frames from an action video taken from three different view for both RGB and Skeletal channels. **Bottom**: Four consecutive frames from an action video from both RGB and Skeletal channels.[5]

[5] Taken from [40]

The final descriptor is learned using a graph convolutional network, which facilitates convolutions over irregular domains structured as arbitrary graphs, in addition to regular structures such as grid graphs formed from image pixels. It benefits the model through enabling it to scale with availability of more training data. We establish the gains in representation power through geometric and kinematic features and consolidate them. Upon this foundation, we build a better action recognition system by adding a well-thought body model and convolution formulations to achieve the culmination of several fragments of persevered thought experiments. Below, we explain the details of our framework.

We first design a system that computes the angular momentum of pose change using the human skeleton videos, analyzes its trajectory and sub-segments the action sequence into parts having smooth changes in the momentum. By sampling from these segments, we aim to summarize the video which is then used to compute a histogram-based descriptor to perform action classification. The results achieved by this system demonstrate the importance of using geometric and kinematic features in understanding human motion using skeleton videos. Through this work, we build a foundation of geometric-kinematic (GK) features.

Recently, manifold-based methods which exploit high-order geometric properties of the data lying on well-studied manifolds like Riemannian and Lie groups have pushed forward the research frontiers in action recognition [41, 42]. Along similar lines, graph convolutions have been proposed by drawing similarities with traditional image-based CNNs [8] as well as by operating in spectral domain and coming back [43, 44]. A human skeleton can be represented as a spatial graph where the joints are the vertices and the natural connections are the edges in the graph. Convolutions can be performed on such a graph to extract meaningful features and perform action classification. Moreover, instead of using 3D joint locations as the features (or signals) at the vertices, GK features can be used to boost recognition performance. Hence, we design a graph convolutional network which uses GK features at the vertices as input, to perform action recognition from S-videos. Inspired by Deformable Part-based Models and the fact that actions can be broken down into movements of individual body parts, we design a part-based human skeleton graph to perform action recognition. We propose a part-based graph convolutional network (PB-GCN) to learn the classification function over the part-based human skeletal graph. We achieve state-of-the-art performance on the benchmark dataset NTURGB+D [13] with a huge margin of improvement and outperform previous methods by a good margin on a small and challenging dataset HDM05 [45]. This demonstrates the effectiveness and scalability of our method, which is attributed to the structure of part-based graph convolutional network (PB-GCN) and using geometric + kinematic signals instead of 3D joint coordinates at the vertices of the skeletal graph.

1.5 Contributions

The contributions of this work are:

- We design a framework for temporal segmentation of trimmed action videos using angular momentum of pose change as a kinematic prior. The temporal segmentation results in meaningful summarization of the skeletal sequence.

- We propose two new histogram-based descriptors: Histogram of Relative Joint Displacements (HORJD) and Histogram of Angular Momentum (HAM) for generating a compact representation of the summarized action video.

- We achieve competitive performance for action recognition on MSRAction3D [11] and UT-Kinect [12] datasets using our segmentation and representation pipeline.

- A detailed discussion on the background about graph convolutional networks, starting with introduction to geometric deep learning and laying the mathematical foundations of the graph convolutions.

- We propose a part-based graph convolutional network for action recognition from skeleton videos, a framework inspired from success of Deformable Part-based Models that can be applied to any graph with known properties.

- We propose to use relative coordinates and temporal displacements instead of 3D joint coordinates, which are the geometric and kinematic signals computed from absolute 3D joint coordinates, at the vertices for learning parameters of GCN. We get great improvements in performance due to them and we explain their reasons.

- We achieve state-of-the-art results for action recognition on two challenging benchmark datasets NTURGB+D [13] and HDM05 [45], outperforming previous methods with a large margin.

1.6 Thesis Organization

The remainder of this thesis is organized as follows:

Chapter 2 begins with a layout of growth in concepts used for recognizing actions and goes on to explain several important methods for human action recognition. We also mention the future course of research that is probable to take place.

Chapter 3 consolidates the foundations necessary for understanding the work in this thesis by laying out the background of important concepts related to graph convolutions used in further chapters. We explain the mathematical and theoretical development of graph convolutions, which are necessary to understand the framework using graph convolutional network.

Chapter 4 explains the importance of geometric-kinematic features through the angular momentum based recognition system. This approach is built on a system that does not use deep learning architectures, as the aim was to design a simple framework that is interpretable and works well.

Chapter 5 explains the usefulness of graph convolutions networks along with the idea of using best of both worlds: deep learning and simple features based on geometry and kinematics. The final model that achieves state-of-the-art performance is explained here.

Chapter 6 concludes the thesis with closing remarks and avenues for future work.

Chapter 2

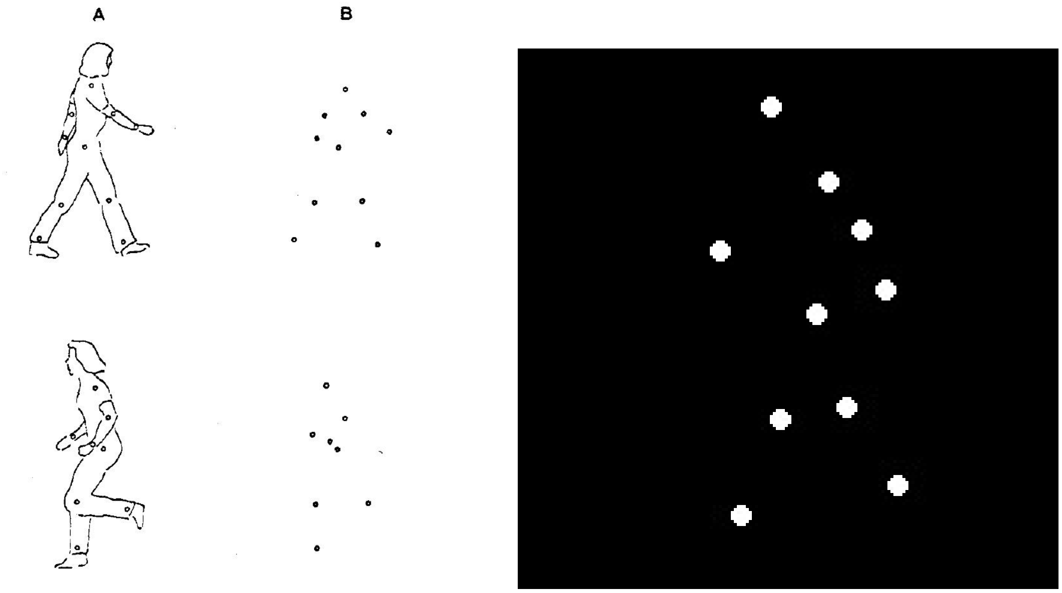
Related Work

Being an important problem in computer vision, human motion analysis for recognition and comprehension of human actions has seen a lot of wisdom being passed on from generation to generation of researchers. Consuming each bit of wisdom passed our way is next to impossible, so we start from ground zero and build an empire of concepts strong enough to hold our heavily complex mechanisms of modern research life, and by extension, our framework for human action recognition from skeleton videos.

2.1 The Wise Old Days

As we explained in Chapter 1, the pioneering work that gave an impetus to formulation of mathematical models for vision-based human action recognition was done by Johansson [22] using the Moving Light Displays (MLD) setup. The genius of that experiment was the observation pointing out the fact that *relative* motion of several 2D points (lights) attached to a human body vividly show human motion characteristics leading to recognition of simple actions. The mere motion of these lights also enabled the recognition of gait and gender of a person [46]. A depiction of the configurations of light displays for two different action poses is shown in Figure 2.1(a) and 2.1(b). In order to emphasize the impression of motion that humans can perceive through the *relative* motion of the light displays, Figure 2.2 shows a subset of stills from the sequence of action “walking”. Indeed, motion information alone is sufficient for perception: a sequence of binary images representing points from a moving object can produce a strong and true-to-life three-dimensional perception.

Johansson *et al.* attempted to explain the interpretation of MLDs in terms of a low-level ”spatio-temporal differentiation and integration” [47]. According to their theory, the peripheral layers of human visual system develop a notion of a hierarchy of coordinate systems that assist in understanding of motion patterns based on simple vector analysis. **Johansson suggests that the choice of the coordinate system of a point in motion is dependent on the two-dimensional velocity of that point, which is an important cue for interpretation of motion** [47]. They, however, face difficulties in defining the



(a)

(b)

Figure 2.1: Demonstration of how the configurations of the moving light displays look like.[1]

[1] (a) Taken from [22] and (b) Source URL

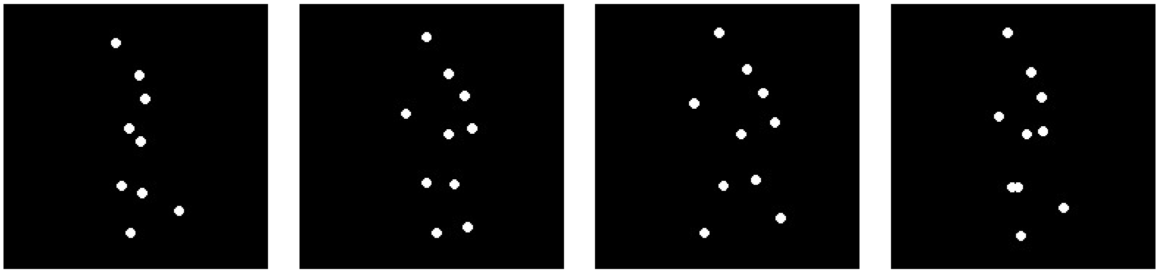


Figure 2.2: A sequence of frames taken from “walking” action. The progression of stills show how the *relative* motion of the light displays induce sensory information about human motion.[2]

[2] Source URL

suitable coordinate system hierarchy through rule-based modeling. Despite this fact, Johansson *et al.* presented a large body of evidence for the hypothesis about human visual system performing a kind of vector decomposition in the analysis of MLDs. We use the 3D displacement of the joints per unit time, or the temporal displacements, which is a kinematic prior that has a good impact on understanding of human motion as explained by Johansson.

Human skeletal data in actions videos have similarities with MLD data in that the location of several joints are provided. Skeletal data is easier to interpret from the point of view of actions as the connections between joints are also well-defined. From the two theories of interpretation of MLD data [24], using motion information directly for recognition is used in modern action recognition systems using human skeletal data extract from RGBD videos. Motion-based recognition systems consist of recognition of objects or motions directly from motion information extracted from the sequence of images. The two main steps in this approach consist of [24]:

- Finding suitable representations of motions based on the motion cues and organizing them into useful compact representations, from which models are created as necessary.
- Matching some unknown input with the model and perform pattern classification using classification techniques.

This is the wisdom that has been conserved over a long time of research in motion interpretation of humans and is the higher level description of the framework that we use in our action recognition approaches.

2.2 The Progression Until Today

We briefly discuss the methods in the literature of action recognition over the past 20 years to build a timeline of how the field has progressed and what is to be learned for the realization of the works presented in this manuscript. Previous works in recognition of human actions can be broadly classified into the type of input information used: (1) RGB videos, (2) RGBD videos and (3) Skeletal videos. We journey through the related works for action recognition using RGB and skeletal videos, and establish a foundation of ideas.

2.2.1 Action Recognition from RGB Videos

One of the earliest works in action recognition using still images was done by Bobick and Davis [1]. They introduce the notion of Motion Energy Image (MEI) and Motion History Image (MHI), whose underlying idea is to encode the motion-related information present in a single image. The MHI and MEI for an action sequence is shown in Figure 2.3. The idea of MEI was extended in Blank *et al.* [2], where they represent an action using a 3D shape induced from the actor's silhouette in space and time. Such a 3D shape induced from silhouette is shown in Figure 2.4.

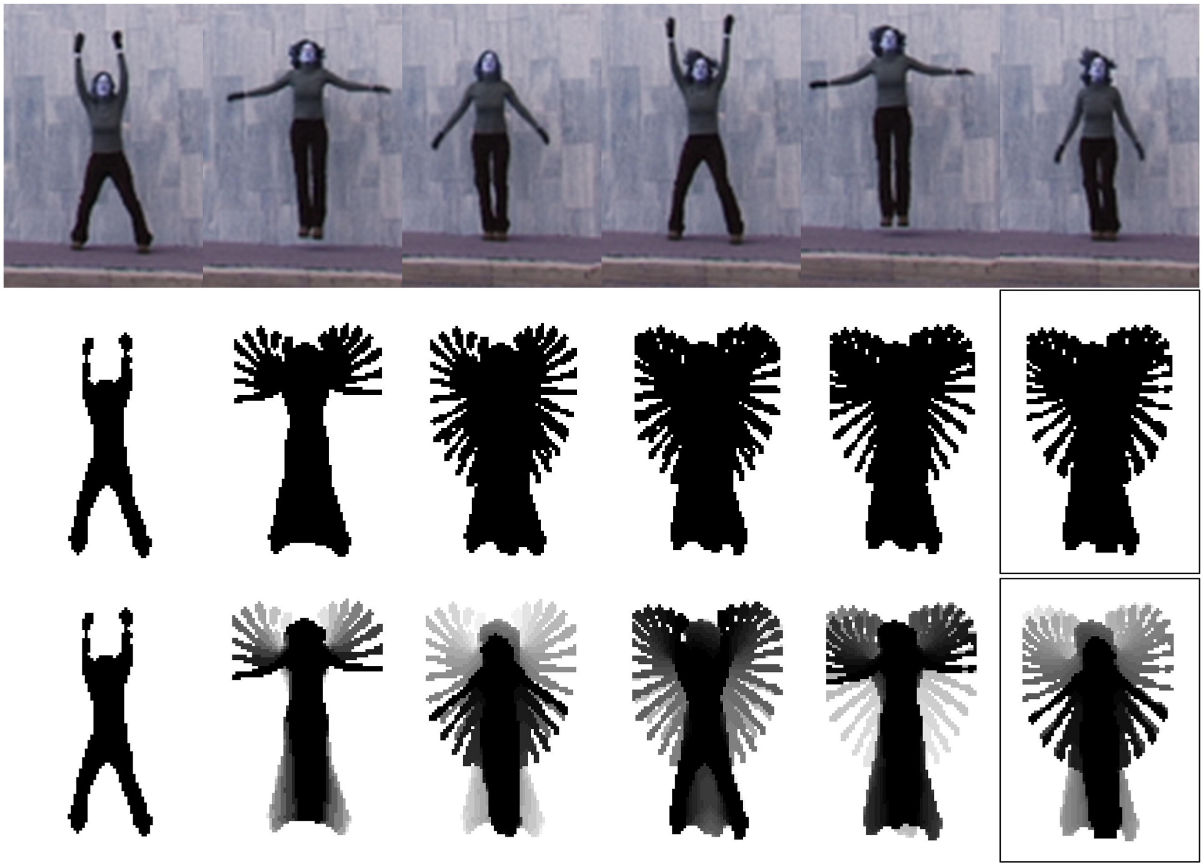


Figure 2.3: **Top**: Image stills from an action sequence of jumping and waving. **Middle**: Motion Energy Images for the action sequence [1]. **Bottom**: Motion History Images for the action sequence [1].[3]

[3] Taken from [38]

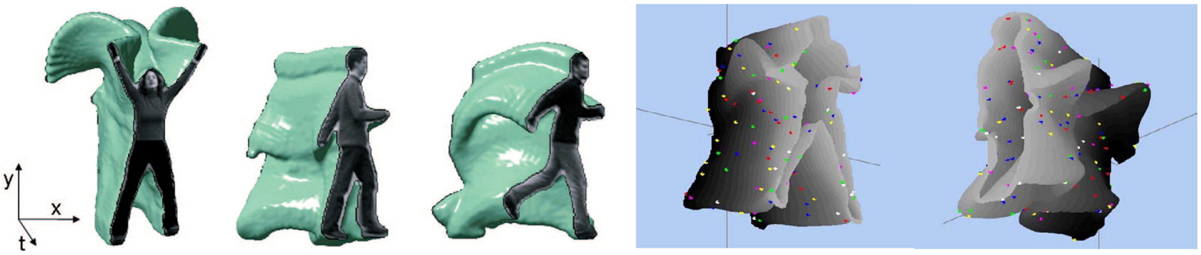


Figure 2.4: **Left**: The spatio-temporal volumes used by Blank *et al.* [2] to describe the evolution of an action. The 3D representation is converted to a 2D map by computing the average time taken by a point to reach the boundary. **Right**: The spatio-temporal surfaces of Yilmaz and Shah [3] for a tennis serve and a walking sequence. The surface geometry (e.g., peaks, valleys) is used to characterize the action.

Changes in direction, speed and shape of an *Space-Time Volume* (STV) inherently characterize the underlying action. Action sketch is a set of properties extracted from the surface of a STV (e.g., Gaussian curvature) and is shown to be robust to view point changes. This is an important concept as we use the notion of geometric and kinematic signals in our approaches based on similar ideas about the changes in direction and speed of the human skeletons.

The representations explained above are very rigid in their structure and fail to capture the complexity of action dynamics which are attributed to point away from the sillhouttes [48]. Hence, local and deep representations became popular. The seminal work on local representations based on Space-Time Interest Points (STIPs) by Laptev [49] inspired further research in this area. In order to extract motion information from the videos, optical flow is computed which provides pixel level motion in X and Y directions. Laptev *et al.* [50] propose to use a Histogram of Optical Flow (HoF), which was extended by Dalal *et al.* [51] to Motion Boundary Histogram (MBH), which encodes the spatial derivative of optical flow fields. An illustration of X and Y motion from optical flow, with the MB image in shown in Figure 2.5. It is significant to note that spatial and temporal derivatives for pixel level motion are helpful in providing the action recognition performance, as similar concepts when applied to skeletal videos also show good performance statistics.

Initial efforts in pushing the frontiers of action recognition performance coming at the end of local representations era was based on dense trajectories. Wang *et al.* [52] propose that sparse interest points do not handle motions effectively and hence they propose to use dense trajectories to compute statistical descriptors for action recognition. The pipeline for dense trajectory computation is shown in Figure 2.6. Transitioning into era of Convolutional Neural Networks (CNNs), the area of action recognition saw a huge boost in the performance due to use of deep learning frameworks involving CNNs due to their powerful feature extraction capabilities. Two-stream spatio-temporal networks [53], spatio-temporal residual networks [54] and temporal segment networks [55] took charge of reaching human level recognition accuracies and have been successful to a large extent. The most important idea to take away from these approaches is the use of trajectories and optical flow, which are derivatives of the information at pixel level.

2.2.2 Action Recognition from Skeletal Videos

Skeletal data consists of locations of human skeleton joints, which are extracted either using MOCAP or sensor-based devices such as Kinect. Skeletons estimated from MOCAP systems are robust to illumination and viewpoint variations, but skeletal data from depth and RGB images are often noisy and error prone due to occlusions and viewpoint variations.

Early methods for recognition of actions from skeletal videos used differential physics-based quantities for the spatio-temporal representation of actions. Zanfir *et al.* [56] propose a descriptor called *Moving Pose*, which encodes the speed and acceleration of a pose centered in a window of a fixed length that is moved over the entire video to calculate the quantities. Several works that followed used higher level quantities such as change in acceleration or used kinetic energy to segment the S-videos and cal-

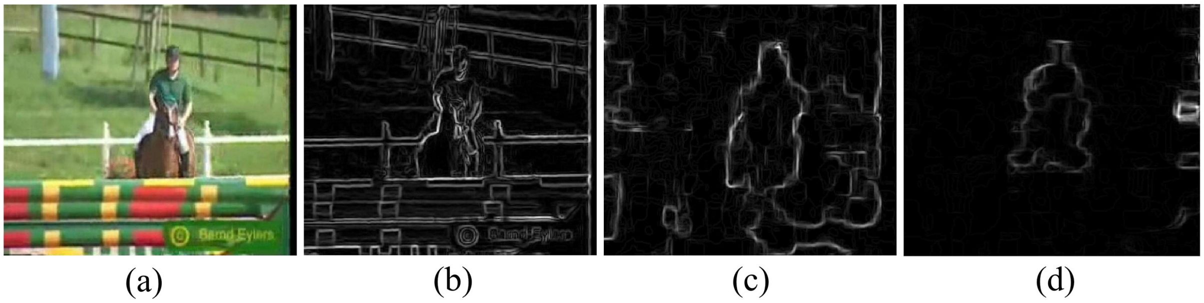


Figure 2.5: Illustration of the optical flow and Motion Boundary images. (b) horizontal component of the optical flow, (c) vertical component of optical flow, (d) Motion boundaries. Spatial gradients disrgard motion information but the Motion Boundaries capture the boundaries at which motion is taking place and hence encode motion better. Motion Boundary Histogram is the gradient of the optical flow fields.[4]

[4] Taken from [38]

culate a descriptor for action recognition [57]. These methods are closely related to our approach for action segmentation and representation using angular momentum.

However, it is intuitive to consider the videos as a sequence of features with each feature extracted from the individual frames in the videos. Based on this, LSTM-based methods are the most obvious choice for learning from such data. Du *et al.* [4] propose a five part-based representation of the human skeleton, which is fed to a hierarchical bi-directional RNN having a separate branch for each of the parts. This is illustrated in Figure 2.7. Shahroudy *et al.* [13] propose a part-aware LSTM architecture wherein memory is split across part-based cells. It is argued that keeping the context of each body part independent and representing the output of the P-LSTM unit as a combination of independent body part context information is more efficient. These methods aim to model the temporal evolution of the skeletal data for recognizing actions but the spatial information is also important in discriminating between action classes. To exploit this dependency, Liu *et al.* [6] proposed a spatio-temporal LSTM (ST-LSTM) network which extends learning to both temporal and spatial domains. ST-LSTM explicitly models the dependencies between the joints and applies recurrent analysis over spatial and temporal domains concurrently. This was extended by Liu *et al.* [5] where they proposed a Global Context-Aware Attention LSTM (GCA-LSTM) (illustrated in Figure 2.8) to selectively focus on the informative joints in the action sequence with the assistance of global context information. Song *et al.* [58] proposed an end-to-end spatio-temporal attention model with LSTM to automatically learn the discriminative joints in each frame and the importance of joints along the temporal axis. Differently from previous works that adopted the coordinates of joints as input, Zhang *et al.* [59] investigated a set of simple geometric features of skeleton using 3-layer LSTM framework, which shows the usefulness of geometric information fed as input compared to absolute joint coordinates. We use ideas from part-based frameworks and application of geometric signals present in previous works as they have shown to benefit performance without increasing the model complexity.

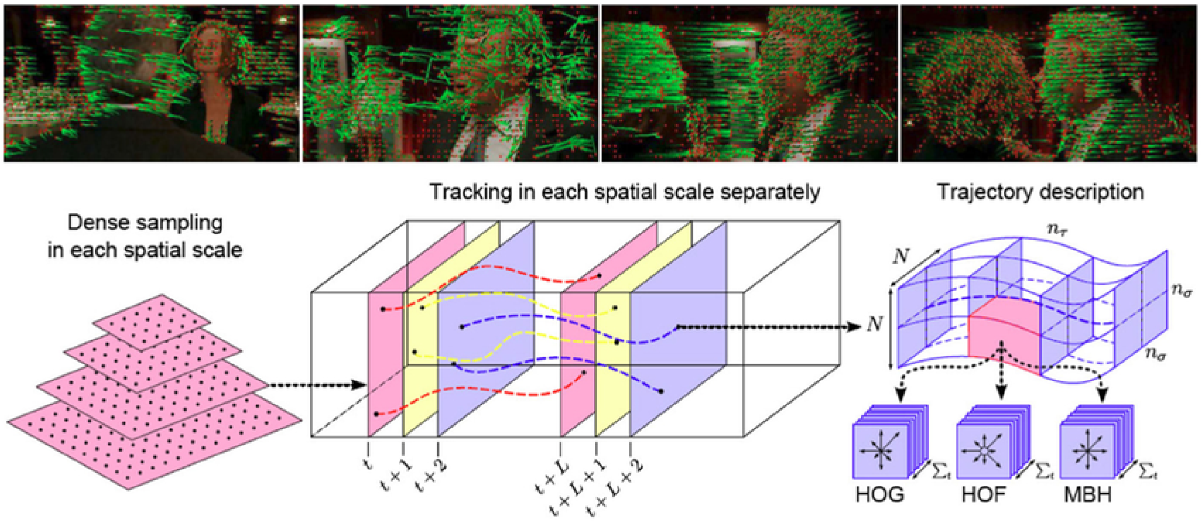


Figure 2.6: **Left**: Feature points are sampled densely for multiple spatial scales. **Middle**: Tracking is performed in the corresponding spatial scale over L frames. **Right**: Trajectory descriptors are based on its shape represented by relative point coordinates as well as appearance and motion information over a local neighborhood of $N \times N$ pixels along the trajectory. In order to capture the structure information, the trajectory neighborhood is divided into a spatio-temporal grid of size $n\sigma \times n\sigma \times n\tau$.[5]

[5] Taken from [52]

Due to the capability of CNNs to learn powerful features from images, action recognition from skeletal videos has seen frameworks that convert skeletal data to images to encode spatio-temporal information that can be extracted by CNNs. Ke *et al.* [60] convert the skeletons into three gray clips, corresponding to X, Y and Z dimensions, each having four images which are encoded with relative coordinates computed with respect to two shoulders and two hips, respectively. The achieve good recognition accuracies by feeding these clips to a pretrained VGG-19 and classifying the extracted features. Such methods lose interpretability as they encode skeletons into images and hence interpreting the filters learned is very difficult. On the other hand, Kim and Rieter [61] propose a temporal convolutional network having a Resnet backbone, to retain interpretability and learn a deep action recognition model. We aim at achieving both the qualities in our approaches and therefore, strive to keep our framework simple, interpretable as well as deep.

2.3 The Future Beckons

Recently, there has been a huge explosion in the field of geometric deep learning due to the capability to design convolution-like functions on non-euclidean domains and in particular, the arbitrary graph-based structures. These methods have shown potential in various core problems of computer vision such as 3D shape correspondence and analysis [62], document analysis [63] and and skeleton-based

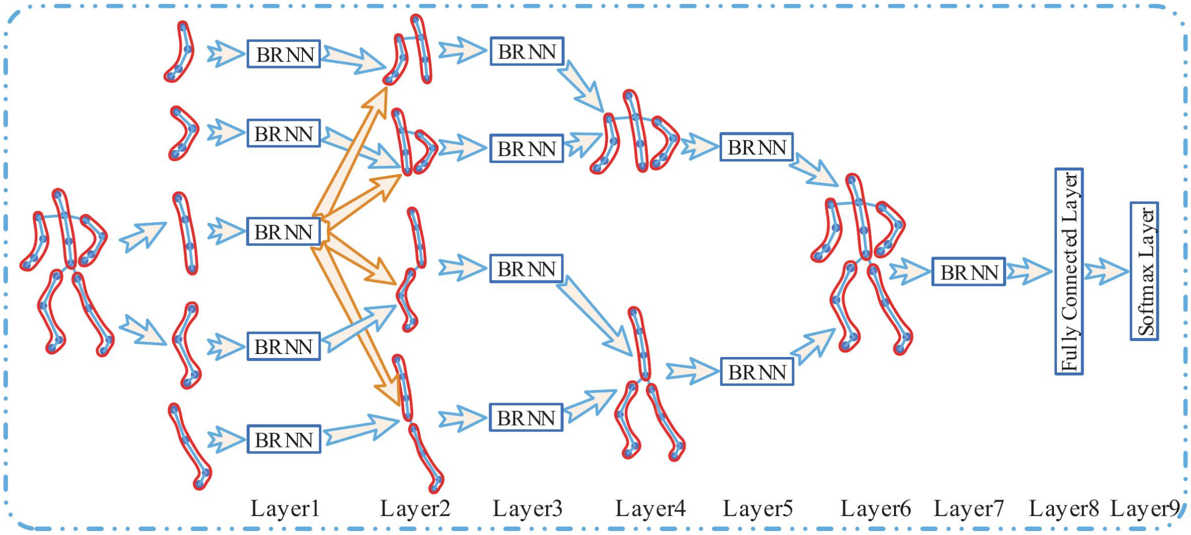


Figure 2.7: Architecture of the hierarchical RNN of Du *et al.* [4].

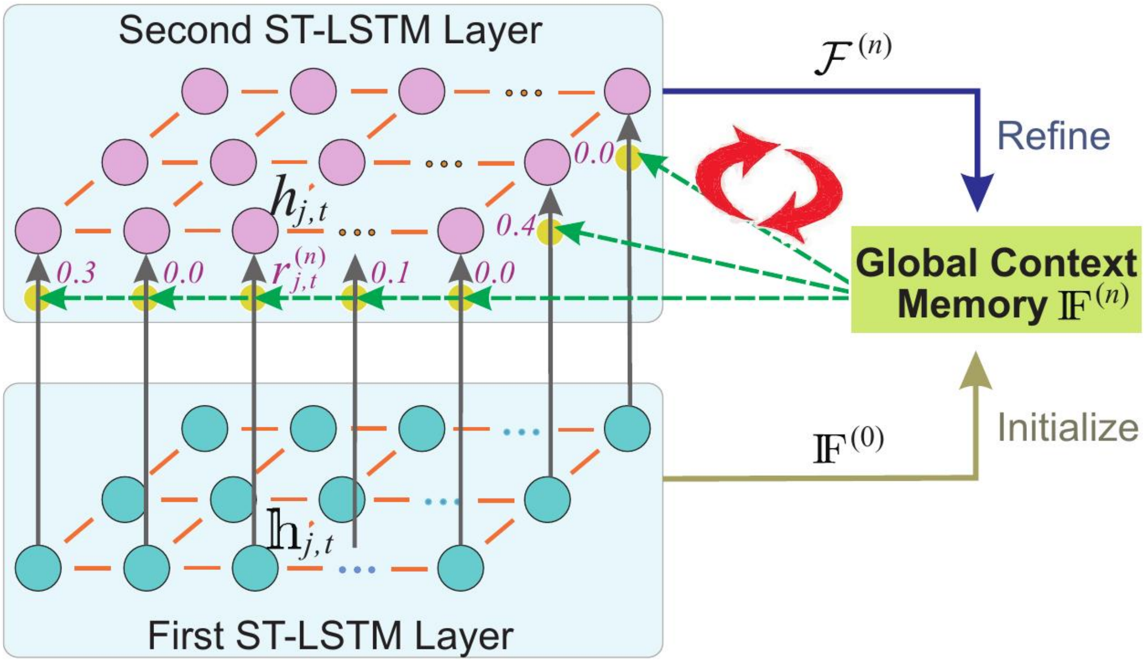


Figure 2.8: Architecture of the GCA-LSTM of Liu *et al.* [5], which uses ST-LSTM of Liu *et al.* [6] as the component LSTMs.

action recognition [9, 64]. It has also shown promise in learning molecular fingerprints [65]. In general, graph convolutional networks (GCNs) have become an extremely popular framework for learning from arbitrary structured graph domains and have shown good capabilities. However, the performance of GCNs is nowhere near the performance of CNNs on implicitly regular-structured image domains (grid-like structure).

It is a matter of time until GCNs catch up and show great potential in their learning prowess. Geometric deep learning is coming and it is the future. We use graph convolutional networks for our framework based on part-based formation of human skeleton graph and perform action recognition using it. We achieve state-of-the-art performance compared to all previous methods for skeletal action recognition on the largest benchmark dataset [13].

Chapter 3

Background: Deep Learning on Graphs

In this chapter, we introduce concepts related to action recognition and geometric models, including graph convolutional networks. Geometric models majorly aim at incorporating structural information to the problem domain for better learning performance of the models. We explain the theory related to them in detail below.

3.1 Introduction: Geometric Deep Learning

Geometric deep learning entails a wide range of techniques aimed at generalizing deep neural models operating on structured domains having euclidean data such as grids, to non-structured (or non-uniform) domains such as graphs and manifolds. Many scientific fields study data with an underlying structure belonging to a non-Euclidean space. Examples of such fields of study include social networks in computational social sciences, communications-based sensor networks, networks to understand functional brain connectivity, regulatory networks in genetics and surfaces involving meshes in computer graphics. Such geometric data have large sizes for most applications (in the case of social networks, on the scale of billions) and exhibit complex properties, which make them a natural target for machine learning techniques. As deep neural networks have demonstrated a great potential recently for a broad range of problems from computer vision, natural language processing, and audio analysis, we would like to use learning techniques similar to them. However, tools from deep learning have been most successful on data with an underlying Euclidean or grid-like structure, and networks that model the input data are designed by incorporating the invariances of these structures.

The success of deep neural networks is attributed to their capability to estimate the local statistics and in turn grasp upon the statistical properties of the data at a stationary location as well as the composition of such properties over several locations, and such statistical properties are exhibited by natural images, video and speech [66, 67]. Physics-based understanding of such statistical properties has been studied [68] and certain class of convolutional neural networks are formalized based on their contributions [69, 70]. As explained in Lecun *et al.* [71], “In image analysis applications, one can consider images as functions on the Euclidean space (plane), sampled on a grid. In this setting, stationarity is owed to shift-invariance, locality is due to the local connectivity, and compositionality stems from the multi-resolution structure of the grid.” Such properties make the use of CNNs very suitable due to the ability to learn local statistics and propagate them hierarchically across the layers, which reduces the number of parameters greatly as well as the inherent presence of certain priors about the data which are fitting, especially for natural images [72, 73].

Due to the attractive properties of CNNs in case of learning problems involving image, speech or video, there has been a growing interest in recent times to apply concepts from CNN-based models to non-Euclidean geometric data. For instance, in social networks, the characteristics of users can be modeled as signals on the vertices of the social graph [74]. Sensor networks are graph models of distributed interconnected sensors, whose readings are modelled as time-dependent signals on the vertices. In genetics, gene expression data are modeled as signals defined on the regulatory network [75]. In neuroscience, graph models are used to represent anatomical and functional structures of the brain. In computer graphics and vision, 3D objects are modeled as Riemannian manifolds (surfaces) endowed with properties such as color texture.

The properties such as global parameterization, well-defined coordinate system and vector space, etc. present in euclidean domain are not applicable to data lying in non-Euclidean domain. As a result, convolutions are not well-defined in the non-Euclidean domain. We explain how key ingredients from CNNs are translated to model non-euclidean data such as skeletal videos for human actions, by introducing the concepts of graph-structured manifolds and convolutions on them.

Geometric learning: Skeletal action recognition Broadly speaking, we can distinguish between two classes of geometric learning problems. In the first class of problems, the goal is to *characterize the structure* of the data. The second class of problems deals with *analyzing functions* defined on a given non-Euclidean domain. We can further break down the problems belonging to second class into two subclasses: problems where the domain is *fixed* and those where *multiple domains* are given. For example, the problem of action recognition from skeletal videos (S-videos) falls under the second class, where we need to learn a *function* defined on the skeletal graph which is our non-Euclidean domain and is *fixed*, in order to perform classification of a given S-video into a predefined set of action classes. Hence, we consider skeletal action recognition as a geometric learning problem where the domain is non-euclidean and we learn a *function* over this domain for supervised classification of actions.

3.2 Where are we: Manifold learning

Roughly, a manifold is a space that is locally Euclidean. For example, consider a 3D sphere such as the shape of the Earth. At every point on the surface of the sphere, one can define a local coordinate system which assigns Euclidean coordinates to the points lying in that system. Representations of skeletal data are used that lie on a certain manifold and learning techniques are applied to identify the actions through the analysis of the action representations lying on the manifolds. Vemulapalli *et* *al.* [41] explicitly model the geometric relationships between different body parts using rotations and translations in 3D space. The resulting representation for the skeletal action sequences lie in the Lie group $SE(3) \times \ldots \times SE(3)$, which is a curved manifold. Huang *et al.* [76] use this representation of skeletal sequences and attempt to develop a deep learning framework that incorporates Lie group structure into a deep network architecture to learn more appropriate Lie group features for 3D action recognition. They design rotation mapping layers to transform the input Lie group features into desirable ones, which are aligned better in the temporal domain. To reduce the high feature dimensionality, the architecture is equipped with rotation pooling layers for the elements on the Lie group. On the other hand, Devanne *et al.* [42] develop a method to perform shape analysis of the 3D joint location trajectories lying on a Riemannian manifold. Huang *et al.* [14] also develop a method to perform Symmetric Positive Definite (SPD) matrix learning on matrices containing visual data, which lie on the Riemannian manifold, and perform classification using the learning technique, which is applied to the task of skeletal action recognition.

These methods have been proven successful in achieving robustness to several noisy inputs and invariance to translation, rotation and scale. Such properties are desirable when performing tasks that involve arbitrary structured data that can contain noisy inputs and benefit greatly from translation and rotation invariance, such as action recognition from skeletal videos. Manifold learning techniques have made a huge impact on how we perceive and represent data, which has given impetus to the development of graph convolutional networks. We explain the necessary background required for understanding formalization of graph convolutions below.

3.3 Where do we go from here: Deep Learning on graphs

Traditional machine learning applications cope with graph structured data by using a preprocessing phase which maps the graph structured information to a simpler representation, *e.g.*vectors of reals. In other words, the preprocessing step first "squashes" the graph structured data into a vector of reals and then deal with the preprocessed data using a list based data processing technique. However, important information, *e.g.*, the topological dependency of information on node n may be lost during the preprocessing stage and the final result may depend, in an unpredictable manner, on the details of the preprocessing algorithm. RNNs and Markov chains are the two techniques that are commonly applied to graph and node focused problems, which perform such a preprocessing. Scarscelli *et al.* [7] were the first ones to develop a formal theory for graph neural networks (GNNs) by extending the concepts of Recursive Neural Networks (RNNs). However, we are interested in the theory behind formulation of graph convolutions rather than a graph neural network which is similar to RNNs.

Niepert *et al.* [8] formulate convolutions on graphs drawing similarities from convolutions on images that operate on locally connected regions of the input. The main aim of convolutions on graphs is to "learn a function on graph-structured data (non-Euclidean domain) which can be used for classification and regression problems on unseen graphs". In the case of skeletal action recognition, the graph con-

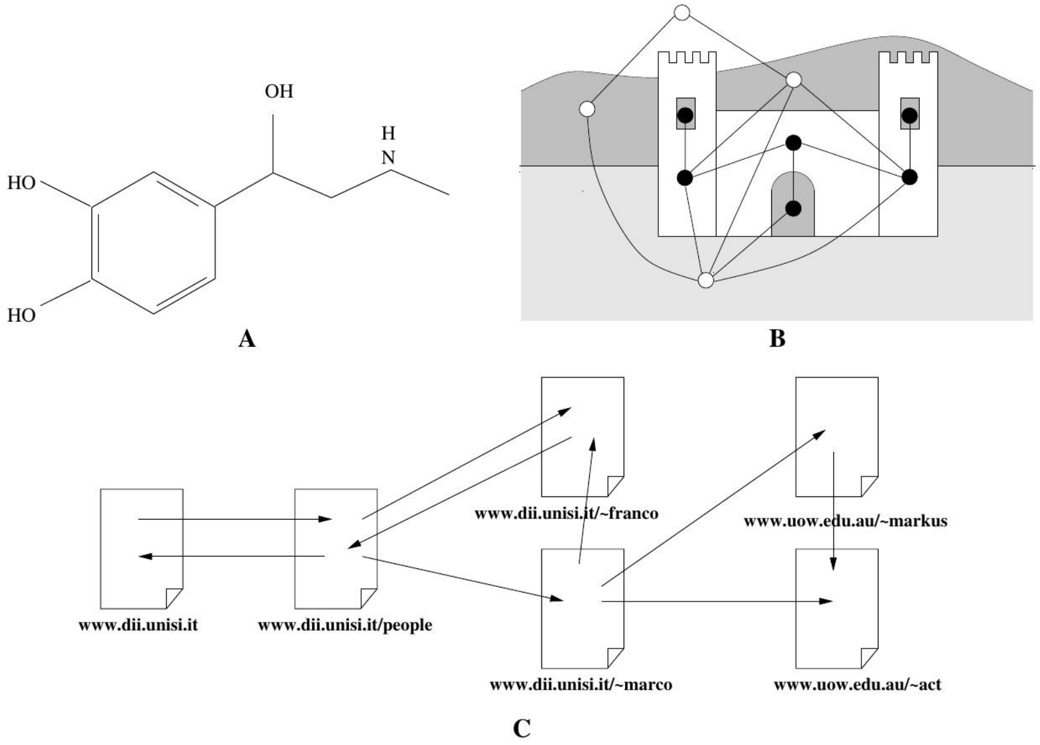


Figure 3.1: Different problems where the data can be represented as graphs [7]. (A) A biological molecule (adrenaline), (B) An image and (C) A subset of documents on the web.

volutions are expected to learn a function that can classify unseen skeletal videos to one of the action classes it has seen in the training set of videos. Figure 3.2 shows the application of a convolution on an image. The image can be represented as a grid with the nodes corresponding to the pixels of the image. The red node in the receptive field of size 3×3 is the root pixel where the convolution is centered. Each of the 3×3 receptive fields create a neighborhood graph for the red root pixels (sequence 1-4 in Figure 3.2). As the pixels in an image have an implicit spatial ordering, the sequence of nodes (root nodes) for which the neighborhood graphs are formed, is uniquely determined. However, in case of arbitrary graphs where there is no spatial, temporal or other sort of ordering among the nodes, two problems need to be solved: (i) Determining the node sequences for which neighborhood graphs are created and (ii) computing a normalization of neighborhood graphs, that is, a unique mapping from a graph representation into a vector space representation.

The pipeline for solving the two problems mentioned above proposed in Niepert *et al.* [8], which we use in our work, is shown in Figure 3.3 and 3.4. Given a collection of graphs, the framework proposed (PATCHY-SAN) does the following: (1) Select a sequence of a fixed number of nodes from the graph vertices, (2) Assemble a fixed-size neighborhood for each node in the selected sequence, (3) Normalize

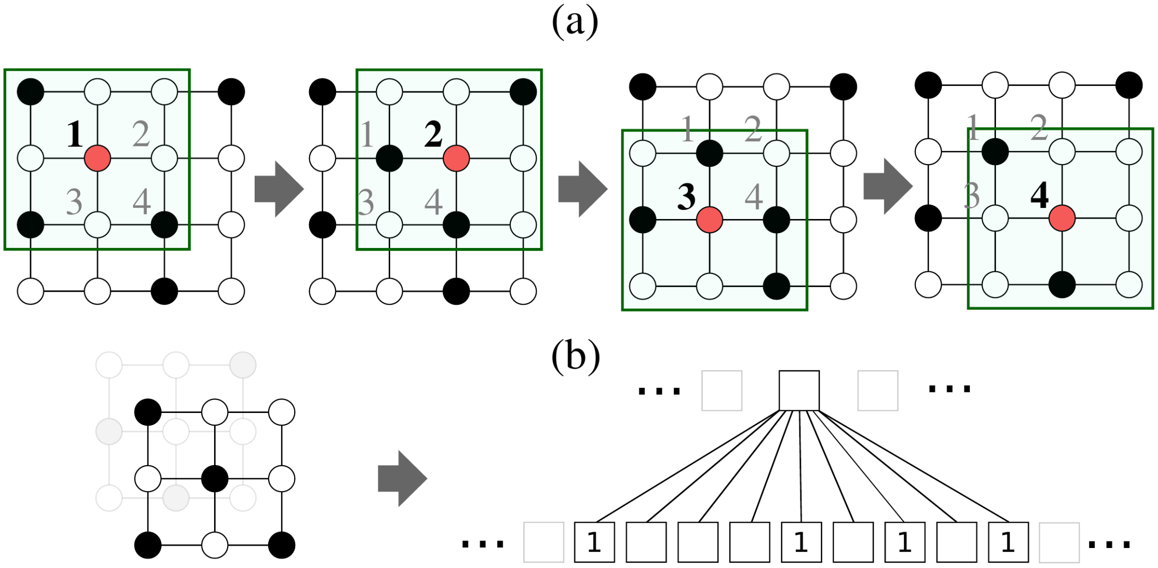


Figure 3.2: Illustration of CNN on images: (a) A CNN with a receptive field of size 3×3. The field is moved over an image from left to right and top to bottom using a particular stride (here: 1) and zero-padding (here: none). (b) The values read by the receptive fields are transformed into a linear layer and fed to a convolutional architecture. The node sequence for which the receptive fields are created and the shapes of the receptive fields, are fully determined by the hyper-parameters [8].

the neighborhood graph assembled for each node and (4) Learn the neighborhood representations using CNNs from the resulting normalized patches.

3.3.1 Notations reference

A graph $G = (V, E)$ has the set of vertices $V = \{v_1, \ldots, v_n\}$ and the set of edges $E \subseteq V \times V$. Let $|V| = n$ and $|E| = m$. Each graph can be represented by an adjacency matrix $\mathbf{A}$ of size $n \times n$, where $\mathbf{A}_{i,j} = 1$ if $(v_i, v_j) \in E$ and $\mathbf{A}_{i,j} = 0$ otherwise. In this case, we say v_i and v_j are adjacent if $\mathbf{A}_{i,j} = 1$ and vertex v_i has position i in $\mathbf{A}$. Node and edge attributes are features that attain one value for each node and edge of a graph. A walk is a sequence of nodes in a graph, in which consecutive nodes are connected by an edge. A path is a walk with distinct nodes. We write $\mathbf{d}(u, v)$ to denote the distance between u and v, viz. the length of the shortest path between u and v. $N_1(v)$ is the 1-neighborhood of a node, which contains all nodes that are adjacent to v.

Labeling and Node Partitions: In order to impose an order on nodes, PATCHY-SAN utilizes graph labelings. A graph labeling l is a function $l : V \rightarrow S$ from the set of vertices V to an ordered set S such as the real numbers and integers. A ranking (or coloring) is a function $\mathbf{r} : V \rightarrow \{1, \ldots, |V|\}$ and every labeling induces a ranking r with $r(u) < r(v)$ if and only if $l(u) > l(v)$. If the labeling l of graph G is injective, it determines a total order of G's vertices and a unique adjacency matrix $\mathbf{A}^l(G)$ of G where

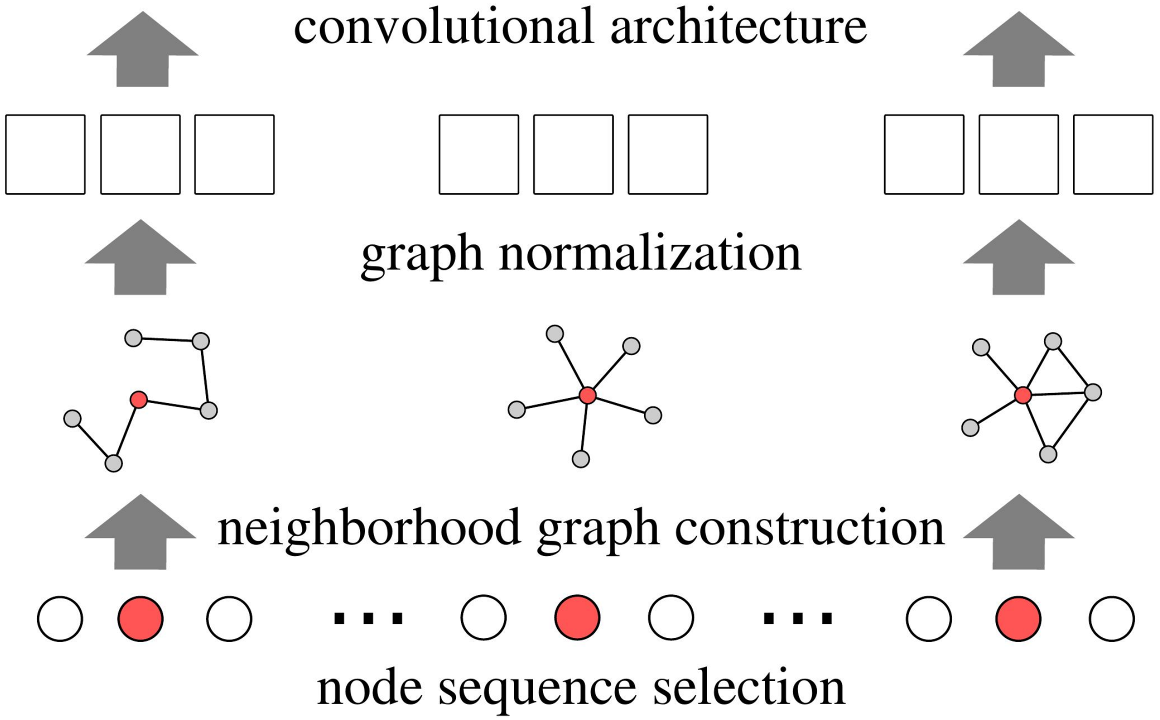


Figure 3.3: An illustration of the Niepert's architecture [8]. A node sequence is selected from a graph via a graph labeling procedure. For some nodes in the sequence, a local neighborhood graph is assembled and normalized. The normalized neighborhoods are used as receptive fields and combined with existing CNN components.

vertex v has position $\mathbf{r}(v)$ in $\mathbf{A}^l(G)$. Moreover, every graph labeling induces a partition $\{V_1, \ldots, V_n\}$ on the set of vertices V with $u, v \in V_i$ if and only if $l(u) = l(v)$.

3.3.2 Convolutions on Graphs

To show the connection between CNNs and PATCHY-SAN, as shown in Figure 3.2(a), the CNNs on images are seen as choosing a sequence of nodes in the square grid graph representing the image and building a normalized neighborhood graph (illustrated in Figure 3.2(b)), which is the receptive field for the chosen node. We now explain the constituent steps in the framework for formalizing convolutions for any arbitrary graph.

SAN → SELECTION OF NODE SEQUENCE: This step determines the nodes in the graph for which we create the neighborhood graphs (similar to choosing an receptive field for convolution) and then define the node ordering in the neighborhood using graph labeling function (in images the receptive fields have an implicit ordering among pixels). Algorithm 1 defines one possible procedure for selection of the sequence of nodes: (1) The nodes are sorted using the graph labeling function. The sorted sequence is traversed with a stride s and each node encountered is chosen and included in the sequence until

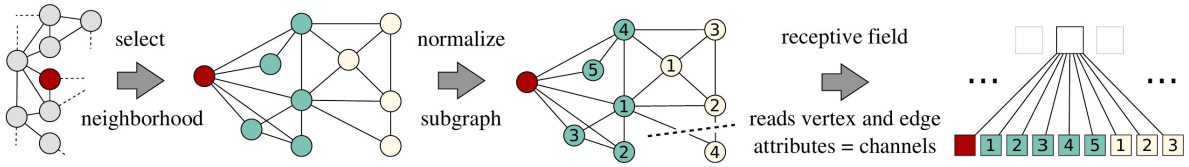


Figure 3.4: The normalization is performed for each of the graphs induced on the neighborhood of a root node v (the red node; node colors indicate distance to the root node). A graph labeling is used to rank the nodes and to create the normalized receptive fields, one of size k (here: $k = 9$) for node attributes and one of size $k \times k$ for edge attributes. Normalization also includes cropping of excess nodes and padding with dummy nodes. Each vertex (edge) attribute corresponds to an input channel with the respective receptive field.[1]

[1] Taken from [8]

the number of required nodes is met or the list of nodes is exhausted, (2) For each of the nodes in the sequence, receptive fields are created to enable performing convolutions on them.

In the case of skeletal action recognition, this step involves selecting all the joints as the sequence of nodes, because the skeletal joints have a well-defined spatial ordering according to the numbering of the joints produced by the motion capture device used. Also, all the joints are important from the point of view of learning the spatial and temporal relationships necessary for characterizing the action sequence and hence all neighborhoods are considered.

```
Algorithm 1: SAN → Select Node Sequence
  Input: graph labeling procedure $l$, graph $G = (V, E)$, stride $s$, width $w$, receptive field size $k$
  $V_{sort}$ = top $w$ elements of $V$ according to $l$;
  $i = 1, j = 1$;
  while $j < w$ do
      if $i \leq |V_{sort}|$ then
          f = ReceptiveField($V_{sort}[i]$);
      else
          f = ZeroReceptiveField();
      end
      apply f to each input channel;
      $i = i + s, j = j + 1$;
  end
```

SAN → Assembly of Neighborhoods: For each node in the selected sequence, a receptive field is created by selecting a neighborhood around the nodes. Algorithm 2 is called by Algorithm 3 to first choose the set of nodes to be included in the root node's neighborhood for creating the receptive field.

```
Algorithm 2: SAN → Assemble the Neighborhood
    Input: vertex $v$, receptive field size $k$
    Output: set of neighborhood nodes $N$ for $v$
    $N = [v]$;
    $L = [v]$;
    while $|N| < k$ and $|L| > 0$ do
        $L = \cup_{v \in L} N_1(v)$;
        $N = N \cup L$;
    end
    return the set of vertices $N$
```

The neighborhood assembly steps involve: (1) Taking the root node v and the receptive field size k as input, perform a breadth-first traversal around v and include the discovered nodes in the neighborhood set N, (2) Increase the distance from v and repeat step 1, (3) If the size of neighborhood set is smaller than k, add the 1-neighborhood of the vertices most recently added to N and continue until $|N| \geq k$, (4) End if there are no more neighbors to add.

Note that, number of neighbors for each node in the sequence may be different, unlike the case in image convolutions (padding can be done to make the number same). For our task of skeletal action recognition, we choose to use 1-neighborhood for each node in the selected sequence of nodes. This is due to the fact that the development of graph convolutions has not matured enough to include higher neighborhoods [77] and is left for future works.

```
Algorithm 3: ReceptiveField → Create the receptive field
    Input: vertex $v$, graph labeling procedure $l$, receptive field size $k$
    $N = $ SAN$(v, k)$;
    $G_{norm} = $ SAN$(N, v, l, k)$;
    return $G_{norm}$
```

SA**N** → Normalization of Graph: The receptive field for a node is constructed by normalizing the neighborhood assembled in the previous step. Illustrated in Figure 3.4, the normalization imposes an order on the nodes of the neighborhood graph so as to map from the unordered graph space to a vector space with a linear order. The basic idea is to leverage graph labeling procedures that assigns nodes of two different graphs to a similar relative position in the respective adjacency matrices if and only if their structural roles within the graphs are similar. This is the problem of *optimal graph normalization* and is considered to be NP-hard [8]. Hence, we perform graph normalization that may assign same label to multiple nodes in the assembled neighborhood and relaxing the restriction for optimal normalization. In our work for skeletal action recognition using graph convolutional networks, we define and explain our labeling functions, which is left to the later chapters.

```
Algorithm 4: SA$\underline{\text{N}}$ → Graph Normalization
  Input: subset of vertices $U$ from original graph $G$, vertex $v$, graph labeling procedure $l$, receptive
      field size $k$
  Output: receptive field for $v$
  compute ranking $\mathbf{r}$ of $U$ using $l$, subject to $\forall\, u, v \in U : \mathrm{d}(u, v) < \mathrm{d}(w, v) \Rightarrow \mathbf{r}(u) < \mathbf{r}(w)$;
  if $|U| > k$ then
      $N$ =top $k$ vertices in $U$ according to $\mathbf{r}$;
      compute ranking $\mathbf{r}$ of $N$ using $l$, subject to $\forall\, u, v \in N : \mathrm{d}(u, v) < \mathrm{d}(w, v) \Rightarrow \mathbf{r}(u) < \mathbf{r}(w)$;
  else
      if $|V| < k$ then
          $N = U$ and $k - |U|$ dummy nodes
      else
          $N = U$
      end
  end
  construct subgraph $G[N]$ for the vertices $N$ canonicalize $G[N]$, respecting the prior coloring $\mathbf{r}$
  return $G[N]$
```

APPROXIMATION OF CONVOLUTIONS → PROPAGATION RULE: Kipf and Welling [77] propose a propagation rule for multi-layered Graph Convolutional Networks (GCNs) that we use in our works. The propagation rule is derived as a first-order approximation of the convolutions with support k defined by Defferrard *et al.* [44], who propose spectral graph convolutions that convert the input domain to fourier domain, perform the convolutions and come back to the input domain. The propagation rule is defined as:

$$H^{(l+1)} = \sigma(\mathcal{D}^{-1/2}\mathbf{A}\mathcal{D}^{-1/2}H^{(l)}W^{(l)}) \tag{3.1}$$

where, $H^{(k)}$ is the hidden layer at level k, $\mathcal{D}$ is the diagonal degree matrix computed from the adjacency matrix $\mathbf{A}$ and W is the matrix representing the filter weights.

We use this propagation rule for the convolutions defined in our GCN for skeletal action recognition. This sums up our discussion on the background for graph convolutions, with an attempt to lay out a clear foundation of the concepts required to understand the frameworks discussed later.

Chapter 4

Segmentation and Representation of Skeletal Videos using Angular Momentum

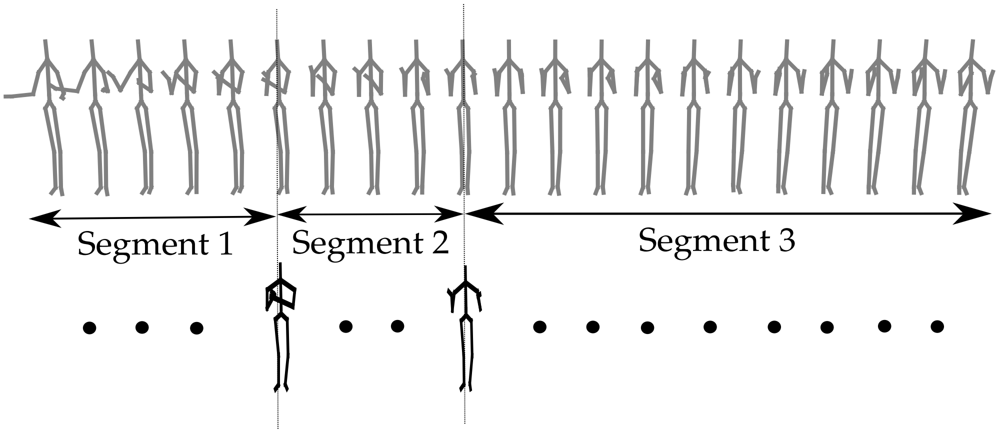


Figure 4.1: Segmentation for trimmed skeletal video of action *clapHands*. The pose sequence of the entire video is shown on top and the segment boundaries are shown below. Poses occurring at the segment boundary capture the extreme poses for the action: “hands at the time of clap” and “hands farthest away”.

In this work, we pursue a hypothesis regarding use of angular momentum as a physical prior to perform temporal segmentation of trimmed action videos. Pose changes in the skeletal action video are characterized using the changes in angular momentum across time. We identify the segments in action video where pose angles change smoothly by analyzing the time series of angular momentum magnitudes. A summarized pose sequence is extracted from the segmented action video by uniformly sampling the temporal segments. We propose a descriptor for generating a compact representation from the summarized pose sequence that captures angular and linear motion across the sequence. The descriptor consists of a concatenation of two fixed-length representations called Histogram of Relative Joint Displacements (HORJD) and Histogram of Angular Momentum (HAM). We use the resultant compact representation for the task of action recognition on small RGBD datasets and show meaningful summarization of the original pose sequences. The performance obtained is competitive to state-of-the-art and our representation is capable of facilitating more applications such as video retrieval, automatic segmentation, etc.

4.1 Introduction

Due to easy accessibility of depth sensors like Kinect, it has been possible to capture scenes with depth information in realtime. With the help of RGBD data captured from such depth sensors, the skeletal configurations of humans in the video can be obtained for each frame. This information is called skeletal video or *S-video* for RGBD videos involving humans. Several tasks such as human action recognition and human 3D modeling can be performed better using the human 3D pose. There has been a rise in datasets for action recognition which contain S-videos recently [11, 12, 13, 78]. 3D pose estimation algorithms enable use of pose information obtained from RGB videos for several tasks such as key frame extraction [79, 80], video segmentation [81] and summarization [82, 83]. We explore the task of segmenting and recognizing actions from trimmed S-videos consisting of human actions.

Human actions can be interpreted on the basis of physics. Each human body part performs angular motion when an action, such as *arm wave*, *tennis serve*, *read a book*, *check your watch*, *walk*, etc. is performed. Each action involves multiple segments of limb movements and segmenting an action video into segments with smooth angular movements can help in its analysis. Shot segmentation of RGB videos [84] can serve as a model to segment skeletal videos, with pose changes playing a key role. We propose a method for action segmentation by identifying the extrema in angular momentum time series of the skeletal videos. Physical quantities like angular momentum are interpretable and useful for this task, as they aim to capture the force applied during angular changes taking place due to motion of limbs.

A segmented video can be used to summarize the action as well as to recognize it. Various representations have been used for information in a skeletal video. Bag of Words (BoW) and statistical encoding techniques are very popular for representing skeletal videos. BoW representations [56, 85, 86, 87] generate a codebook using clustering or sparse coding which is used to find a single code word for each feature vector. Statistical encoding techniques [88, 89, 90] use simple statistics to generate a final feature representation. This approach generates a fixed-length feature representation, irrespective of the number of poses in the sequence. We use a statistical approach for our encoding. A histogram of relative displacement and angular momentum, binned by their orientations and calculated over multiple time segments with a temporal pyramid, is our compact descriptor for an action video.

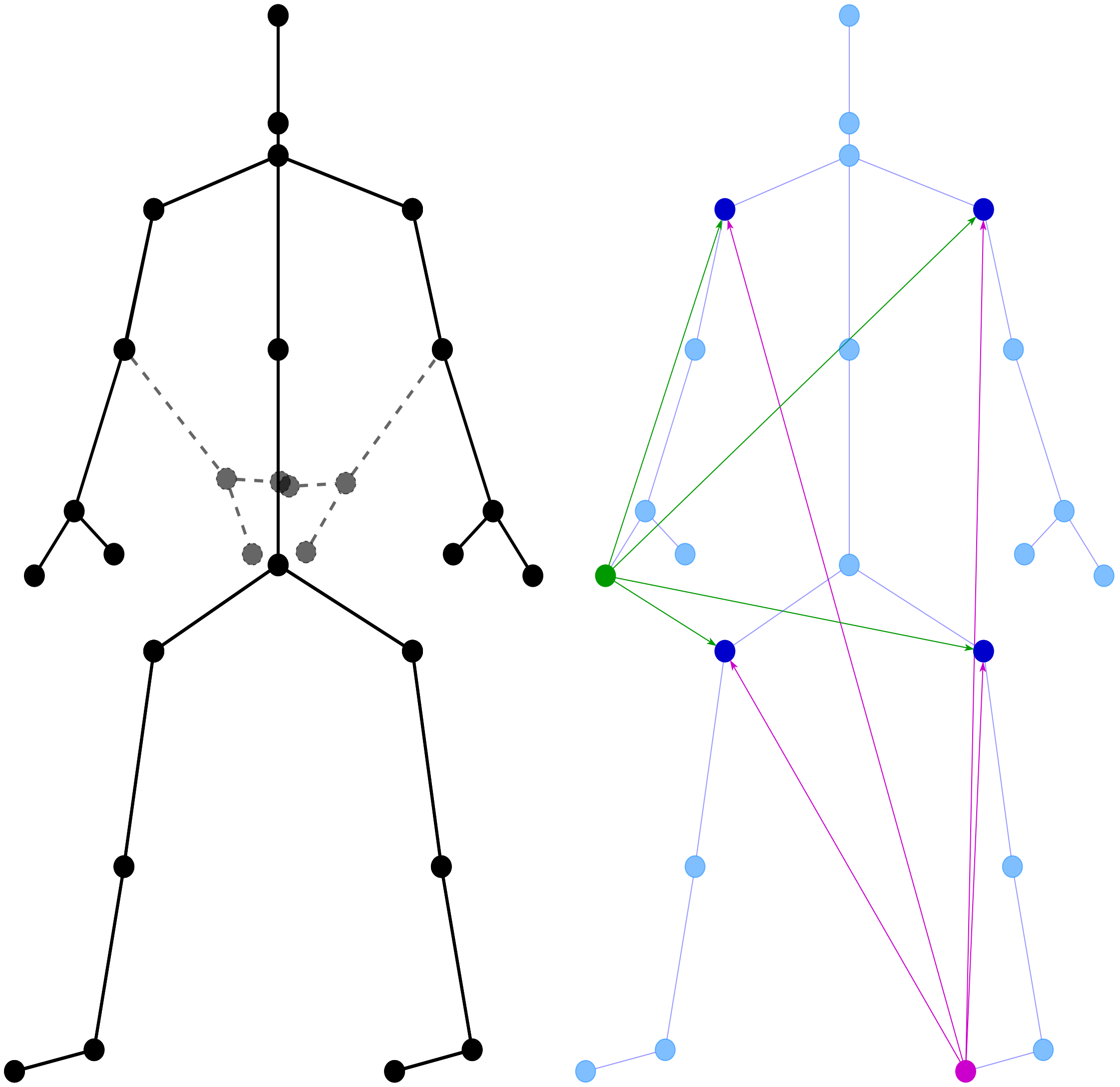

(a) (b)

Figure 4.2: **(a)**: The angular motion of the two hands about their respective elbow joints with the old positions (shown in solid) and new positions (shown in dashed lines), **(b)**: Relative joint vectors calculated for the human skeleton joints with respect to four joints: left shoulder, right shoulder, left hip and right hip. In our method, we use seven joints as the reference set. This image is for illustration of how the relative coordinate vectors look like.

Our method for temporal segmentation of skeletal videos is based on a simple observation that most of the body parts perform circular motion during an action and exhibit change in momentum, which can be quantified using a physical quantity like angular momentum. Segment boundaries are identified by finding the extrema of angular momentum across the video. This has an interpretable meaning: The beginning of an extreme for one action segment and end for the previous can be characterized by detecting time instances where maximum angular motion takes place between poses. This is similar to the concept of scene or shot change used for segmentation in RGB videos. We find the summary of skeletal video by sampling poses uniformly from each action segment. A compact representation for the skeletal video is generated using two proposed descriptors, HAM and HORJD. The major contributions resulting from this work are as follows:

- We propose using angular momentum: an interpretable *kinematic prior* for temporal segmentation of skeletal videos.
- We propose two new histogram based descriptors, namely Histogram of Oriented Relative Joint Displacement (HORJD) and Histogram of Angular Momentum (HAM) for encoding the final pose sequence.
- Applicability to action recognition is demonstrated using quantitative experiments that show the effectiveness of our method.

4.2 Compact Representation of Skeletal Videos

In this section, we explain temporal segmentation of the skeletal video using our angular momentum prior. We will first explain the calculation of angular momentum for a pose change, followed by the segmentation method to identify temporal segment boundaries.

4.2.1 Angular Momentum Prior

Whenever a human performs an action, force is applied to several parts of the human body. Depending on direction of momentum changes, these actions can be of two types: a *stationary* action or a *dynamic* action. In a *stationary* action, the frame of reference of the human is stationary, i.e. the frame of reference has zero velocity at all time instants. Such actions include *brushing*, *drinking water*, *checking watch*, *brushing hair*, etc. In case of a *dynamic* action, the frame of reference of the human is moving with certain velocity which may not be constant. Such actions are more complex due to larger variations possible in motion of the frame of reference and in temporal extents of the action. *Dynamic* actions include *jumping*, *running*, *walking*, etc. For both the classes of actions, circular motions are inevitable due to motion of limbs about a pivot joint. Hence, we use angular momentum to capture the complex dynamics for human actions.

The angular momentum of any body rotating about a center is given by the equation:

$$\overrightarrow{\mathcal{L}} = \overrightarrow{\mathbf{r}} \times \overrightarrow{\mathbf{p}} \tag{4.1}$$

where $\overrightarrow{\mathbf{r}}$ is radius of the body from axis passing through the center and $\overrightarrow{\mathbf{p}}$ is the linear momentum. For a skeleton, we consider each joint as a body with unit mass, with radius equal to the limb length and center being the joint it is rotating about. Hence in our case the angular momentum is defined for an endpoint of each limb of the body. The change in position of arms which contribute to the change in angular momentum are shown in Figure 4.2(a). We explain the process of calculating angular momentum below.

Step 1: We first find the instantaneous linear velocity for each joint during a pose change in the skeletal video. This displacement per unit time (which we refer to as instantaneous velocity) can be calculated using a simple derivative:

$$\forall j \quad \overrightarrow{\mathcal{V}}_{\mathbf{j}}^{\mathbf{\Delta t}} = \left(\frac{\overrightarrow{\mathcal{P}}_{\mathbf{j}}^{\mathbf{t+1}} - \overrightarrow{\mathcal{P}}_{\mathbf{j}}^{\mathbf{t}}}{\mathbf{\Delta t}} \right) \tag{4.2}$$

where, j represents the joint index and t represents the time instance. For calculating the angular momentum of the joint about it's pivot joint, we need the normalized limb vector joining those two joints.

Step 2: Assume $\overrightarrow{\mathcal{P}}_1^t$ and $\overrightarrow{\mathcal{P}}_2^t$ are the 3D joint locations of two adjacent joints (say, the right elbow and right hand) at time t and $\overrightarrow{\mathcal{P}}_1^{t+1}$ and $\overrightarrow{\mathcal{P}}_2^{t+1}$ at time $t+1$. Here, hand joint is moving and elbow joint is the pivot. Using the calculated instantaneous velocity of the hand joint using equation (2) and the normalized limb vector, which is represented by $\hat{\mathbf{r}} = \overrightarrow{\mathbf{r}}/\|\overrightarrow{\mathbf{r}}\|$, the angular momentum equation can be written as follows (the body is considered to have unit mass):

$$\forall j \quad \overrightarrow{\mathcal{L}}_{\mathbf{j}}^{\mathbf{\Delta t}} = \hat{\mathbf{r}} \times \overrightarrow{\mathcal{V}}_{\mathbf{j}}^{\mathbf{\Delta t}} \tag{4.3}$$

Step 3: The magnitude of angular momentum vectors calculated in the previous step is added to get the total angular momentum of the pose change using the following equation:

$$\mathcal{L}^{\mathbf{\Delta t}} = \sum_{\mathbf{j}} \|\overrightarrow{\mathcal{L}}_{\mathbf{j}}^{\mathbf{\Delta t}}\| \tag{4.4}$$

We get a time series of length $t-1$ that represents the change in angular momentum of the poses with time. For example, the angular momentum time series for actions *carry* and *sit down* is shown in Figure x and y respectively. The plot for *carry* contains a lot of peaks due to noise, which we filter out using mean thresholding and non maximum suppression that we explain in the next section.

4.2.2 Motion Segmentation

After calculating the angular momentum time series, we need to find the motion boundaries which exhibit a large change in pose based on angular motion. Before seeking such a boundary in the trajectory, we perform non-maximum suppression [91] to remove unnecessary noise due to noisy joint locations. Non-maximum suppression is commonly used to filter out noisy edge detections in images. We perform

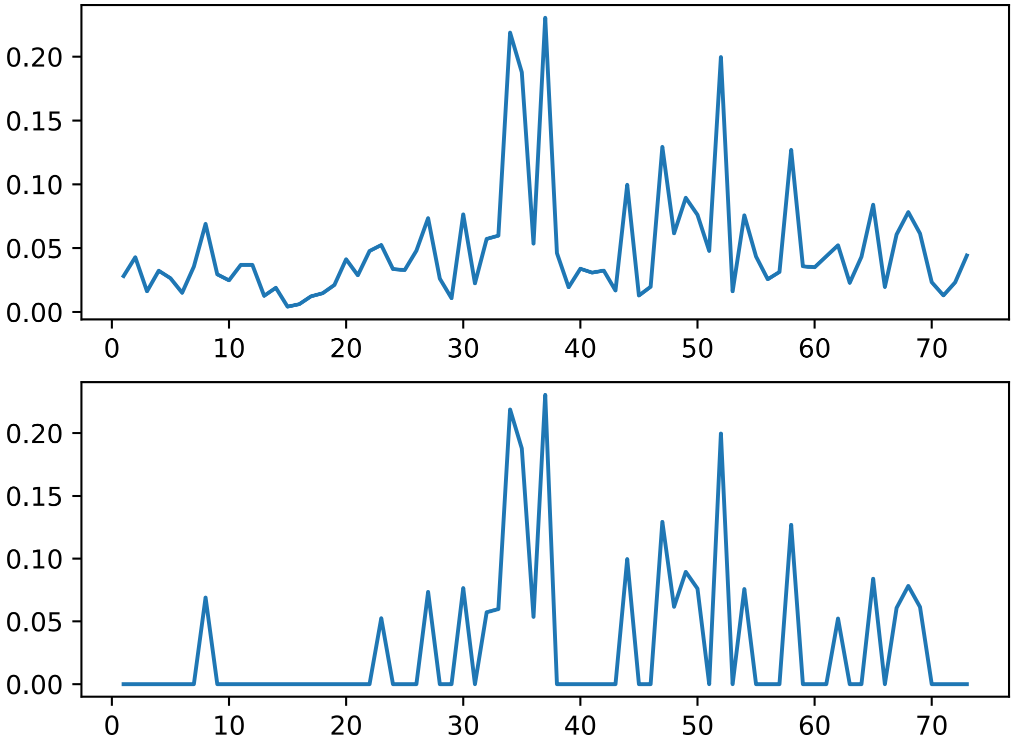


Figure 4.3: Plots for angular momentum trajectory for the action *carry*. X-axis represents the time instants and Y-axis represents the angular momentum magnitudes. **Top**: Shows values without thresholding with mean of the distribution. **Bottom**: Plot obtained after thresholding. Nature of the top plot is due to the fact that action contains a lot of noise in the measurements. This is handled using the thresholded plot and performing non-maximum suppression to find segment boundaries.

a simple 3-neighborhood non-maximum suppression on the time series as follows:

$$\mathbf{slope}(\mathbf{t}) = \| \overrightarrow{\mathbf{L}}^{\mathbf{t}} \| - \| \overrightarrow{\mathbf{L}}^{\mathbf{t-1}} \| \tag{4.5}$$

$$\mathbf{slope}(\mathbf{t+1}) = \| \overrightarrow{\mathbf{L}}^{\mathbf{t+1}} \| - \| \overrightarrow{\mathbf{L}}^{\mathbf{t}} \| \tag{4.6}$$

$$\mathbf{boundary}(\mathbf{t}) = \begin{cases} 1 & \text{for} \quad sign(\mathbf{slope}(\mathbf{t})) \neq \\ & \qquad sign(\mathbf{slope}(\mathbf{t+1})) \\ & \qquad \mathbf{and} \\ & \qquad sign(\mathbf{slope}(\mathbf{t})) > 0 \\ 0 & \text{for} \quad \mathbf{othercases} \end{cases} \tag{4.7}$$

The length of temporal segments may vary across the video. A longer segment implies a gradual change in pose between the motion boundaries whereas a shorter segment implies a swift change in pose. To

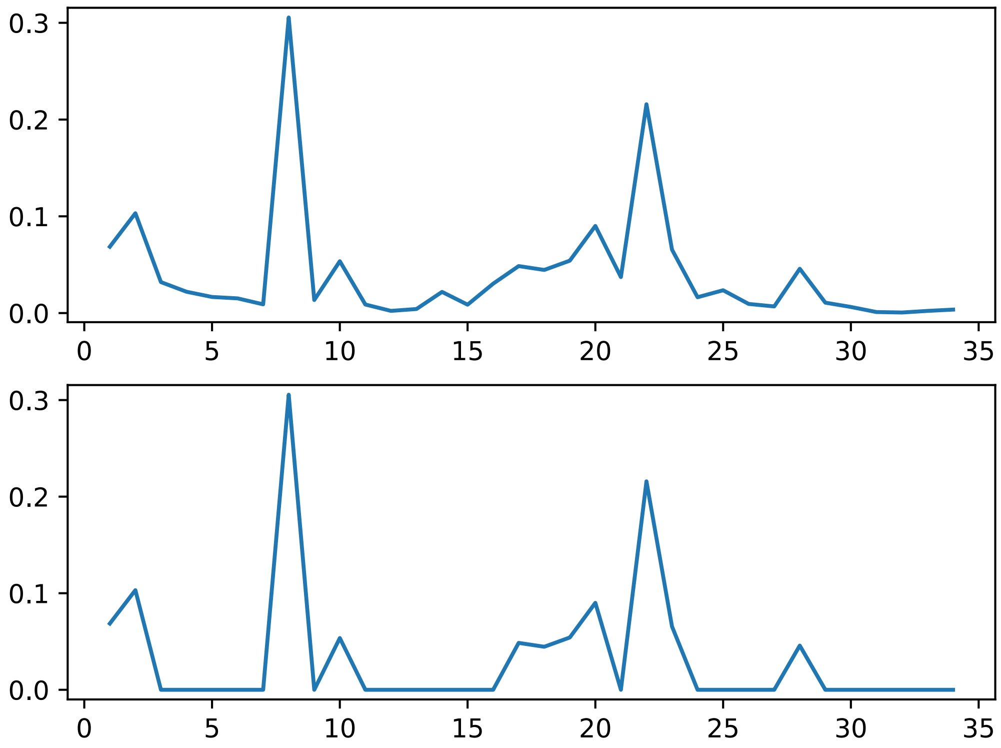


Figure 4.4: Plots for angular momentum trajectory for the action *sit down*. X-axis represents the time instants and Y-axis represents the angular momentum magnitudes. **Top**: Shows values without thresholding with mean of the distribution. **Bottom**: Plot obtained after thresholding. Here, the noise is significantly lower than *carry*.

summarize the action, spacing between poses in longer segments should be more than that in smaller segments. Hence, after finding the temporal motion boundaries for the skeletal sequence, we sample a fixed percentage of equally spaced poses from each of the segments. We fixed the percentage of poses to be 20% as it gave a reasonable sampling frequency and did not suffer from over-sampling or under-sampling in our experiments. For segments having length less than *eight* frames and greater than *two*, we choose a fixed number of samples from that segment. For example, this gives us a summary of the actions as shown in Figure 4.7. Such a summary can be used to generate a GIF after finding the RGB frames corresponding to the poses.

4.2.3 Representation Encoding

For converting our final pose sequence into a fixed length compact representation, we propose two histogram based descriptors called Histogram of Oriented Relative Joint Displacements (HORJD) and

Histogram of Angular Momentum (HAM). We use a temporal pyramid shown in Figure 4.5 to encode action evolution in time. Specifically, we divide the pose sequence into four parts for which the descriptors are computed. Temporally adjacent encodings are added to get two, which are in turn added to get one global descriptor as shown in Figure 4.5. Descriptors at all levels are concatenated to get the 3-level temporal pyramid descriptor for the pose sequence.

4.2.4 HORJD: Histogram of Oriented Relative Joint Displacements

Relative joint locations are more discriminative as compared to absolute joint locations [92]. Motivated by this observation, we propose to use a histogram based statistical encoding for relative joint displacements which represents linear motion in the final pose sequence.

Extremities Sets: For calculating the descriptor we have chosen two sets of joints with respect to which the relative joint locations are found. The first set includes *four* joints: *two* shoulder and *two* hip joints. They are considered to remain stable for most of the actions [93] and hence can model the relative motion of other joints effectively. The second set includes *seven* joints: *two* elbow, *two* hand, the head joint and *two* foot joints. The second set is chosen based on the fact that end point joints in human body tend to be most discriminative for actions that humans perform [94]. Using these two sets, we calculate the relative joint coordinates for four joints in set one with respect to each joint in set two. This gives us $4 \times 7 = 28$ relative joint locations for each pose.

The computation of descriptors involve several steps. We briefly explain the steps as the method is similar to the one proposed in [89]. However, we compute our descriptors on the summarized pose sequence and do not use the entire skeletal video.

Step 1: For each time step Δt, the displacement for each of the 28 relative joint vectors is calculated. Assume that a relative joint vector belonging to a pose at time t is given by (x_t, y_t, z_t). The displacement vector for the relative joint between two adjacent poses in the summary can be represented as $(\Delta x, \Delta y, \Delta z)$ where $\Delta x = (x_{t+1} - x_t)$. Figure 4.2(b) shows the relative joint coordinates vectors for two joints with respect to four reference joints. This step encodes motion information about the action through displacement.

Step 2: This displacement vector is projected on to *xy, yz, xz* planes and the orientation of each 2D vector is calculated. For example, the orientation in *xy* plane is given by $\theta = \arctan(\frac{\Delta y}{\Delta x})$. Orientation is used to decide the bin number for the displacement vector in the histogram.

Step 3: After calculating the displacement vectors and projecting them, their magnitude is added to a bin in the histogram based on the orientation of vector in 2D. The number of bins in histogram is predefined. This is performed for each of the 28 relative joint locations using the three projected vectors giving a histogram of size $28 \times 3 \times n(bins)$. This is the size of descriptor for each pose adjacent change in the summarized video.

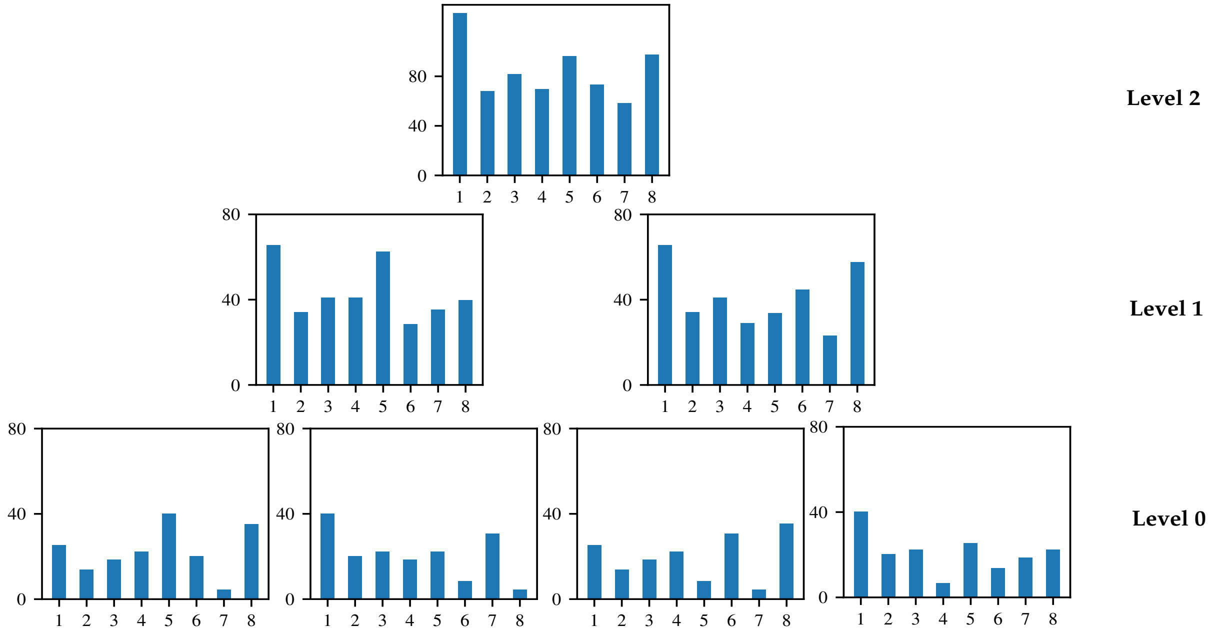


Figure 4.5: Three level temporal pyramid representation for histograms used in our descriptors. Three levels equal a total of seven histograms and a concatenation of these seven descriptors gives the final descriptor.

4.2.5 HAM: Histogram of Angular Momentum

Using our intuition about significance of angular motion in human actions which can be characterized by angular momentum, we propose this descriptor.

Limb Representation: Consider the skeleton representation shown in Figure 4.2(a). Each of the vectors is a limb of the human body which has the potential of performing angular motion about a pivot joint. For calculating our HAM descriptor, we first calculate the 3D vectors joining two adjacent joints from the absolute joint locations while ignoring the torso, as it is unlikely that any angular motion will happen there.

Step 1: The first step is to calculate the angular momentum for a joint (considering it as a body with unit mass) generated due to it's angular motion. We use the equations mentioned in section 3.1 to calculate momentum for each joint and get the final vector $\overrightarrow{\mathbf{L}}_{\mathbf{j}}^{\mathbf{\Delta t}}$ calculated for that Δt (here, $\Delta t = 1$). However, this is done only for the summarized pose sequence in contrast to doing it for entire sequence as shown in section 3.1.

Step 2: After calculating angular momentum vectors, the procedure is exactly same as HORJD descriptor. Projection of vectors followed by binning based on orientation in 2D is used to compute the histogram.

4.3 Experiments and Results

We show the applicability of our method using experiments demonstrating summarization of poses and action classification.

4.3.1 Datasets

Datasets used to show applicability to action classification are listed below:
MSRAction3D [11] This dataset was captured using a depth sensor similar to the Kinect device. The dataset contains 20 action classes and 10 subjects performing the actions. Each subject performs each action 2 or 3 times. There are 567 depth maps in total out of which 10 are either all zeros or too erroneous. This dataset is challenging due to it's high intra-class variation.
UT-Kinect [12] This dataset was recorded using a stationary Kinect. It is very challenging due to the noise in the captured skeleton sequences and variations in viewpoints. It contains 10 action classes and 10 subjects perform each action twice. Total of 199 action sequences are present in the dataset. The skeletons contain 3D locations of 20 joints.

4.3.2 Experimental Setup

We perform the task of action recognition using the summarized pose sequence in order to evaluate the effectiveness of our method quantitatively. We extract HORJD and HAM from the poses and perform action recognition using a linear SVM with $\mathbf{C} = 10.0$ and $\gamma = 0.05$. The results are shown in Table 4.1 and 4.2. For each of the listed experiments, we use 3-level temporal pyramid and show the results. The number of bins for histograms are set to 10 and 12 for HORJD and HAM respectively, which were decided empirically through experimentation. We perform tests with several different ways of extracting the final descriptors (SBS = Segment Boundary Sequence and SS = Summarized Sequence).

We follow two evaluation protocols widely followed for MSRAction3D. In the first protocol, the actions are divided into three action sets [11] and cross subject tests are performed using subjects 1, 3, 5, 7, 9 for training. Average accuracy over the three action segments is reported. In the second protocol, the entire dataset is used to perform the cross subject tests using subjects 1, 3, 5, 7, 9 for training. We report the performance for both as it is not clear in some previous works which approach has been used. For UT-Kinect we follow the cross subject protocol where subjects 2, 4, 6, 8, 10 are used for training and the remaining are used for testing.

4.4 Discussion

What do segment boundary poses signify? The summarized pose sequences for *horizontal arm wave*, *wave hands* and *tennis serve* are shown in Figure 4.7. The segment boundary poses signify the angular motion extremes observed in the actions. For example, in action *wave hands*, the poses in *bold* have captured the extreme where the hands cross each other as well as the extreme where hands are at the widest and farthest from each other. Similar observation can be made for the action of *horizontal arm wave* where poses in *bold* have captured hands at the two extremes of the horizontal wave gestures. For the action of *tennis serve* as well, similar observations can be made. The segment boundary poses have captured the extremes of hand motion using which the serve is being done.

This significance of the segment boundary poses is important as this makes the segmentation of actions more intuitive and interpretable. With good action segments, final pose summaries become more natural and informative.

Why uniform sampling? Sampling a fixed percentage of frames from each segment which are equally spaced ensures that the final summaries are temporally normalized across different instances of the same action. For example, if an action is performed slower in one instance compared to another instance, our method will ensure that the evolution of action does not look different when the summary is observed. This is clearly not a very good choice from recognition point of view, but it provides good summarization as shown in Figure 4.7. The pose sequences we extract should allow a human to identify the action as if he were looking at the original pose sequence.

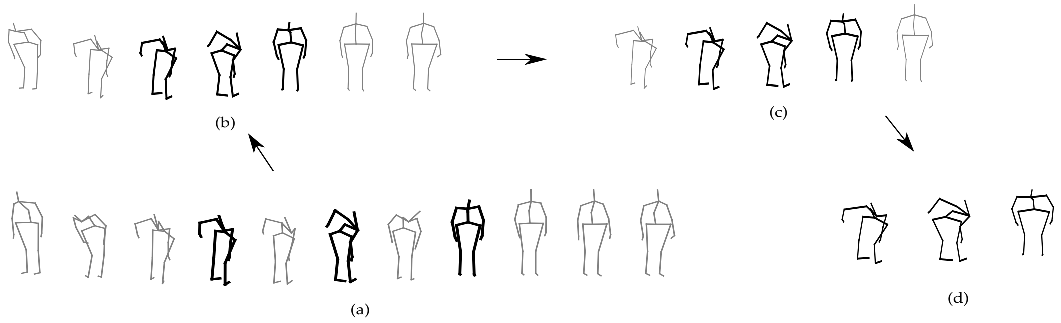


Figure 4.6: (Following the arrows): Effect of choosing different percentage of poses to be extracted from each segment for summarization. This transition goes through 20%, 10%, 5% sampling (parts (a), (b) and (c)) and then finally the segment boundaries itself (part (d)). This implies the required amount of smoothness can be achieved by the sampling frequency.

Performance of HAM + HORJD Based on observations from the recognition rates of different baselines, HAM descriptor is not very effective for MSRAction3D dataset but provides a small boost in performance when combined with HORJD. However, HAM works extremely well for UT-Kinect, even outperforming HORJD while using both SBS and SS. HAM and HORJD together outperform several previous works as they complement each other.

Intuitively, HAM should handle complex angular motions very well and HORJD should be able to distinguish between different relative joint motions. The accuracies for HAM with SS for the three action segments (AS_1, AS_2, AS_3) [11] were 81.72%, 74.11% and 93.39%, which shows that HAM performs much better for classifying complex actions as the ones in AS_3 action set. Similar observation can be made in case of it's performance in baselines as HAM performs well on UT-Kinect which contains complex actions with view point variations. This is consistent to our intuition for HAM descriptor.

As for HORJD, it performs well in differentiating between similar actions and is robust to intra-class variations. This can be seen from it's superior performance on MSRAction3D compared to HAM, which is because MSRAction3D contains noisy skeletons and introduces large intra-class variations in the actions performed. The accuracies for HORJD with SS for the three action segments (AS_1, AS_2, AS_3) [11] were 91.39%, 90.18% and 92.45%, which shows that HORJD performs well in separating intra-class variations in similar actions. HORJD performs marginally better than HOD and COV3DJ (baseline SS+HORJD+SVM) even when using a smaller set of joints and only summarized pose sequence as opposed to the entire skeletal video. This shows the potential of HORJD descriptor.

Our method outperforms all the methods except Meshry *et al.* [85] on MSRAction3D dataset. We pursue our hypothesis of using geometric and kinematic priors (relative coordinates and angular momentum, in our case). However, their method uses a BoW representation and dictionary learning for action recognition while we propose a physically interpretable *prior* for segmenting a skeletal video which is more intuitive. Through segmentation, we enable several applications like video summarization and action recognition.

For UT-Kinect dataset, we outperform all methods except Vemulapalli *et al.* [98]. Their method in based on modeling human actions on a curved manifold called Lie group, to capture view and scale invariant action evolution. This suffers from lack of physical interpretability inherent in human actions as analysis of such manifolds is not easy. Using our simple and general method, we achieve 95% using only the summarized pose sequences while they use the entire skeletal video to model the curve in Lie Group. Our method also points out an interesting property of action videos: all frames are not necessary for understanding the actions.

Summarization capability Summaries generated for action "bend" from MSRAction3D are shown in Figure 4.6. It is clear from the results that action summaries are meaningful as they have captured the important poses in the sequence. We use 41% of the frames in skeletal video for MSRAction3D and 40% for UT-Kinect on average for summarization, when performing a sampling of 20% from each segment. This can be varied by changing the sampling frequency as shown in Figure 4.6. For sampling frequency of 10% and 5%, we use only 18% and 14.88% of the frames in skeletal video for MSRAction3D whereas sampling below 15% for UT-Kinect gives empty segments (due to very small segments). Most compact summary for the sequences can be obtained using the segment boundaries itself as the video summary. Identifying the action using only the segment boundaries is not that easy but they are reasonably representative of the actions. Using only segment boundary poses as the action summary, we use only 14.6% of the frames in skeletal video for MSRAction3D and 15.2% for UT-Kinect on average.

This shows that our method can be used to summarize videos according to required amount of smoothness based on sampling frequency and can enable playing a video at different speeds on the basis of summaries.

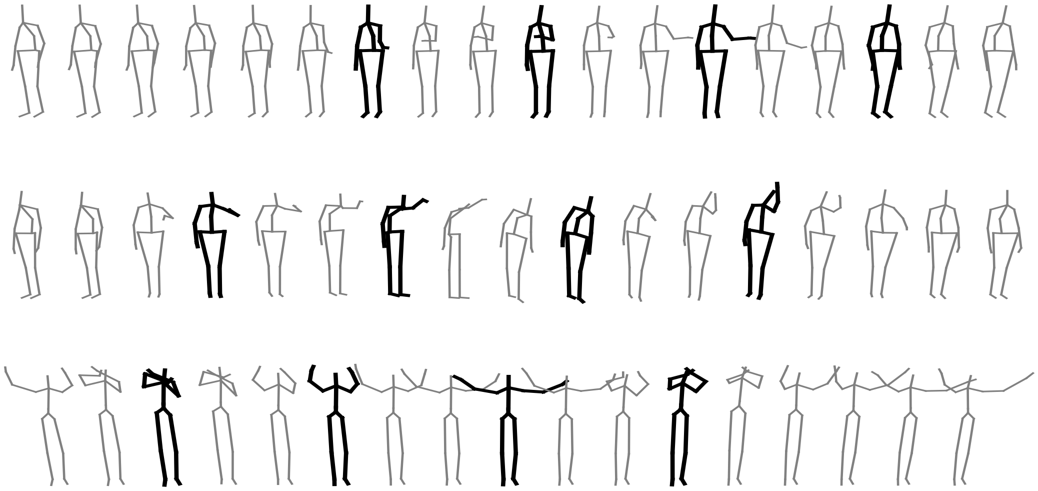


Figure 4.7: (top to bottom): The summarized pose sequences for *horizontal arm wave* and *tennis serve* action from MSRAction3D dataset and *waveHands* action from the UT-Kinect dataset. The sequences show segment boundaries in *bold* which characterize the extreme poses of the actions and the sampling of poses show a smooth transition between these segment boundaries.

4.5 Conclusion

We propose a simple and effective method for temporal segmentation of skeletal videos where we seek to find motion boundaries which represent extremes in any human action. Using angular momentum, we find segments where the pose angles changes smoothly and meaningful summarization is achieved using our segmentation technique. We compute a compact representation for the video using two histogram based features for the summarized poses called Histogram of Relative Joint Displacement (HORJD) and Histogram of Angular Momentum (HAM), computed on the extracted pose summary. Results for action recognition on two challenging datasets demonstrate the effectiveness of our method. Several other applications are possible using our approach, such as video retrieval, gesture recognition, etc.

Our method is dependent on SVMs for classification purposes and computes a spatio-temporal representation of the action based on a non-parametric method. As a result, our framework is unable to scale the performance of recognition with the availability of large amount of training data. Hence, in our next work, we focus on deep learning methods to overcome this limitation on the present framework. However, the present framework has proven the usefulness of geometric + kinematic priors for skeletal action recognition, which is an important contribution of this work. We take this learning further and combine it with the capabilities of deep learning framework based on graph convolutional networks (GCNs). We explain this framework in the next chapter, which achieves state-of-the-art performance for action recognition using skeletal data.

MSRAction3D	
Method	**Accuracy**
Actionlet Ensemble* [95]	88.20
COV3DJ* [90]	90.53
HOD* [89]	91.26
Grassmannian Motion Depth* [96]	86.21
Grassmannian Manifold* [97]	91.21
Moving Pose* [56]	91.70
Lie Group* [98]	89.48
HON4D* [99]	85.85
HON4D + D^*_{disc} [99]	88.89
Bag of 3D points [11]	74.70
HOJ3D [12]	78.98
Joint distances + Key poses [100]	65.71
Bag of Gesturelets [85]	**96.05**
SBS + HORJD + SVM	79.53 / 74.71
SBS + HAM + SVM	80.57 / 73.56
SBS + HORJD + HAM + SVM	81.13 / 80.08
SS + HORJD + SVM	91.34 / 90.42
SS + HAM + SVM	83.07 / 80.08
SS + HORJD + HAM + SVM	**92.55 / 91.18**

Table 4.1: Results on MSRAction3D [11] using cross subject setting in action sets / cross subject setting for entire dataset.

* Use entire videos for action recognition whereas we only use the summarized pose sequences.

UT-Kinect	
Method	**Accuracy**
Grassmannian Manifold* [97]	88.5
HOJ3D [12]	90.92
Informative joints [101]	91.90
Coupled hidden CRFs* [102]	92.00
Lie Group* [98]	**97.08**
Riemannian Manifold* [103]	91.5
Key-pose Motifs [104]	93.5
SBS + HORJD + SVM	82.65
SBS + HAM + SVM	85.71
SBS + HORJD + HAM + SVM	86.73
SS + HORJD + SVM	89.80
SS + HAM + SVM	91.84
SS + HORJD + HAM + SVM	**94.90**

Table 4.2: Results on UT-Kinect [12] for different baselines and comparison to several previous works.

* Use entire videos for action recognition whereas we only use the summarized pose sequences.

Chapter 5

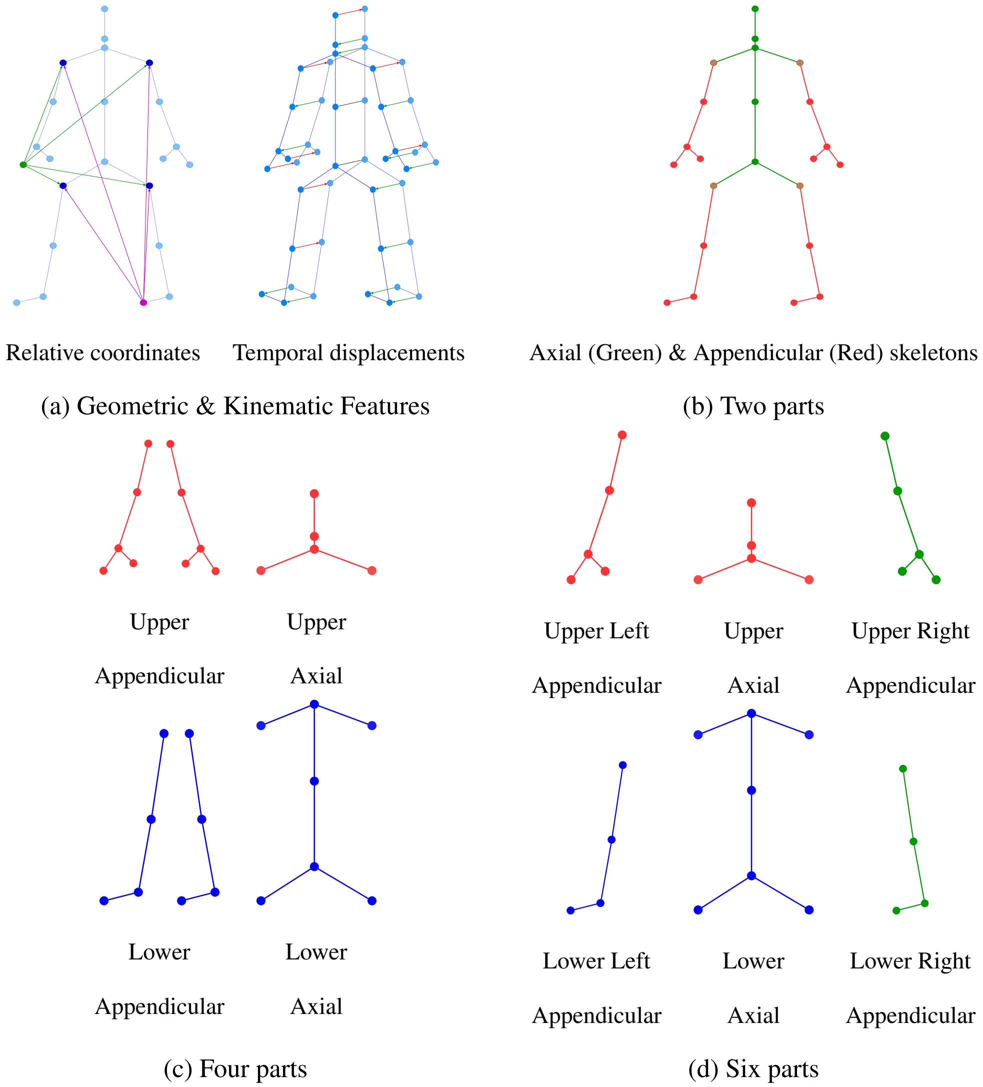
Part-based Graph Convolutional Network for Action Recognition

Human actions comprise of joint motion of articulated body parts or “gestures”. Human skeleton is intuitively represented as a sparse graph with joints as nodes and natural connections between them as edges. Graph convolutional networks have been used to recognize actions from skeletal videos. We introduce a part-based graph convolutional network (PB-GCN) for this task, inspired by Deformable Part-based Models (DPMs). We divide the skeleton graph into four subgraphs with joints shared across them and learn a recognition model using a part-based graph convolutional network. We show that such a model improves performance of recognition, compared to a model using entire skeleton graph. Instead of using 3D joint coordinates as node features, we show that using relative coordinates and temporal displacements boosts performance. Our model achieves state-of-the-art performance on two challenging benchmark datasets NTURGB+D and HDM05, for skeletal action recognition.

5.1 Introduction

Recognizing human actions in videos is necessary for understanding them. Video modalities such as RGB, depth and skeleton provide different types of information for understanding human actions. The S-video (or Skeletal modality) provides 3D joint locations, which is a relatively high level information compared to RGB or depth. With the release of several multi-modal datasets [13, 105, 106], action recognition from S-video has gained significant traction recently [5, 6, 58, 59, 60].

Graph convolutions [8, 44, 77] have been used to learn high level features from arbitrary graph structure. State-of-the-art action recognition from S-videos [9, 64] use graph convolutions, wherein the whole skeleton is treated as a single graph. It is, however, natural to think of human skeleton as a combination of multiple body parts. A body-part based representation can learn the importance of each part and their relations across space and time. We present a model using part-based graph convolutional



(a) Geometric & Kinematic Features

(b) Two parts

(c) Four parts

(d) Six parts

Figure 5.1: **(a)** Geometric and Kinematic features, **(b)** Appendicular and axial body parts: two parts, **(c)** Dividing the appendicular and axial skeletons into upper and lower parts: four parts, **(d)** Dividing appendicular upper and lower skeletons into left and right: six parts.

network for recognizing actions from S-videos, using a novel part-based graph convolution scheme. The model attains better performance for recognition than a model entire skeleton as a single graph. Current models for skeletal action recognition [9, 64] use 3D coordinates as features at each vertex. Geometric features such as relative joint coordinates and motion features such as temporal displacements can be more informative for action recognition. Optical flow helps in action recognition from RGB videos [55] and Manhattan line map helps in generating 3D layout from single image [107]. Geometric feature [59] and kinematic features [108] have been used for skeletal action recognition before. Inspired by these observations, we use a geometric feature that encodes relative joint coordinates and motion feature that encodes temporal displacements at each vertex in our part-based graph convolution model to significant impact.

The major contributions of this work are: (i) Formulation of a general part-based graph convolutional network (PB-GCN) which can be learned for any graph with well-known properties and its application to recognize actions from S-videos, (ii) Use of geometric and motion features in place of 3D joint locations at each vertex to boost recognition performance, and (iii) Exceeding the state-of-the-art on challenging benchmark datasets NTURGB+D and HDM05. The overview of our representation and signals is shown in Figure 5.1.

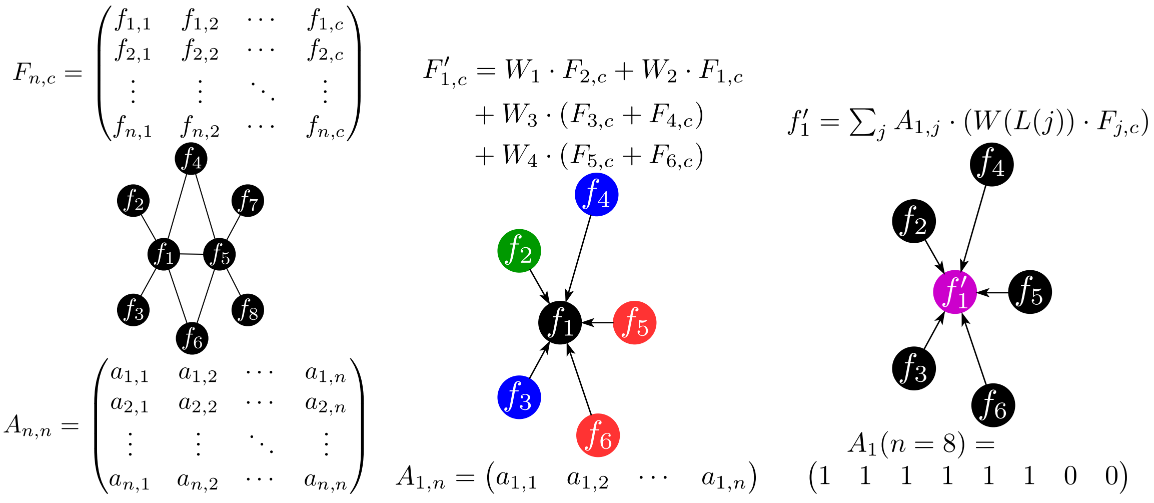
5.2 Previous Methods

5.2.1 Non graph-based methods

Skeletal action recognition has been approached using techniques such as handcrafted feature encodings, complex LSTM networks, image encodings with pretrained-CNNs and manifold-based non-euclidean methods. Non-deep learning methods worked well initially and proved usefulness of several extracted information from S-videos such as joint angles [109], distances [12] and kinematic features [108]. These methods learn from hand designed features using shallow models which do not model spatio-temporal properties of actions very well and constrain learning capacity.

On the other hand, LSTM-based methods were used because S-videos can be thought of as time sequences of features. Spatio-temporal LSTMs [5, 6], attention-based LSTM [58] and simple LSTM networks with part-based skeleton representation [110, 111] have been used. These methods either use complex LSTM models which have to be trained very carefully or use part-based representation with a simple LSTM model. We propose a part-based graph convolutional network that has good learning capacity and uses a part-based representation, inheriting the good qualities of both types of aforementioned approaches. Image encodings of skeletons were proposed to facilitate usage of Imagenet pretrained CNNs to extract spatio-temporal features. Ke *et al.* [60] generate images using relative coordinates while Du *et al.* [112] and Li *et al.* [113] proposed a body part-based image encoding. Due to inherent differences in information in such image encodings and RGB images, it is almost impossible to interpret the learned filters. In contrast, our method is intuitive as it uses a graph-based representation for human skeleton.

Manifold learning techniques have been used for skeletal action recognition, where actions are represented as curves on Lie groups [41] and Riemannian manifold [114]. Deep learning on these manifolds is difficult [76] while deep learning on graphs (also a manifold) has developed recently [44, 77]. Our method uses a human skeleton graph and learns a model using part-based graph convolutional network, exploiting the benefits of deep learning on graphs.


$F_{n,c} = \begin{pmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,c} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,c} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n,1} & f_{n,2} & \cdots & f_{n,c} \end{pmatrix}$
f_1
f_2
f_3
f_4
f_5
f_6
f_7
f_8
$A_{n,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix}$
$F'_{1,c} = W_1 \cdot F_{2,c} + W_2 \cdot F_{1,c} + W_3 \cdot (F_{3,c} + F_{4,c}) + W_4 \cdot (F_{5,c} + F_{6,c})$
$A_{1,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \end{pmatrix}$
$f'_1 = \sum_j A_{1,j} \cdot (W(L(j)) \cdot F_{j,c})$
f'_1
$A_1(n=8) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$

(a) Graph feature and adjacency matrices

(b) Convolution for receptive field of a chosen root vertex f_1

(c) Final convolution equation

Figure 5.2: Equation-based formulation and illustration of a graph convolution

5.2.2 Graph-based methods

Representing S-videos as skeleton graph sequences for recognizing actions had not been explored until recently. Li and Leung [115] construct graphs using a statistical variance measure dependent on joint distances and match them for recognition. Recently, Yan *et al.* [9] and Li *et al.* [64] proposed a spatio-temporal graph convolutional network for action recognition from S-videos. Both the methods construct graphs where the human skeleton is treated as a single graph. Our formulation explores a partitioned skeleton graph with a part-based graph convolutional network and we show that it improves recognition performance. Also, we use relative coordinates and temporal displacements as features at each vertex instead of 3D joint coordinates (see Figure 5.1(a)) which improves action recognition performance.

5.3 Background

A graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of vertices and $\mathcal{E} \subseteq (\mathcal{V} \times \mathcal{V})$ is the set of edges. $\mathbf{A}$ is the graph adjacency matrix having $\mathbf{A}(i,j) = w$, $w \in \mathbb{R} \setminus \{0\}$ if $(v_i, v_j) \in \mathcal{E}$ and $\mathbf{A}(i,j) = 0$ otherwise. $\mathcal{N}_k : v \rightarrow \mathcal{V}$ defines the set of vertices $\mathcal{V}$ in k-neighborhood of v which includes neighbors having shortest path length atmost k from vertex v. A labeling function $\mathbf{L} : \mathcal{V} \rightarrow \{0, 1, \ldots, \mathcal{L} - 1\}$ assigns a label to each vertex in a vertex set $\mathcal{V}$, where $\mathcal{L}$ is the number of unique labels. The adjacency matrix is normalized using a degree matrix as:

$$\mathcal{D}(i,i) = \sum_j \mathbf{A}(i,j) \tag{5.1}$$

$$\mathbf{A}^{\mathbf{norm}} = \mathcal{D}^{-1/2}\mathbf{A}\mathcal{D}^{-1/2} \tag{5.2}$$

Graph convolutions can be formulated using spectral graph theory [44] or spatial convolution [8] on graphs. We focus on spatial convolutions in this paper as they resemble convolutions on regular grid graphs like RGB images [8]. A graph CNN can then be formed by stacking multiple graph convolution units. Graph convolution (shown in Figure 5.2) can be defined as [8]:

$$\mathbf{Y}(v_i) = \sum_{v_j \in \mathcal{N}_k(v_i)} \mathbf{W}(\mathbf{L}(v_j))\mathbf{X}(v_j) \tag{5.3}$$

where, v_i is the root vertex at which the convolution is centered (like center pixel in an image convolution), $\mathbf{W}(\cdot)$ is a filter weight vector of size of $\mathcal{L}$ indexed by the label assigned to neighbor v_j in the k-neighborhood $\mathcal{N}_k(v_i)$, $\mathbf{X}(v_j)$ is the input feature at v_j and $\mathbf{Y}(v_i)$ is the convolved output feature at root vertex v_i. Equation 5.3 can be written in terms of adjacency matrix as:

$$\mathbf{Y}(v_i) = \sum_j \mathbf{A}^{\mathbf{norm}}(i,j)\ \mathbf{W}(\mathbf{L}(v_j))\ \mathbf{X}(v_j) \tag{5.4}$$

$\mathbf{A}^{\mathbf{norm}}(i,j)$ basically defines the neighbors at distance 1 and hence, Equation 5.3 captures a more general form of convolution by using k-order neighborhood $\mathcal{N}_k(v_i)$.

5.3.1 Part-based Graph

Graphs representing real world manifolds can often be thought of as being made up of several parts. For instance, a graph representing a complex molecule consists of several simple structures, such as structure of a protein biomolecule, which can be divided into several polypeptide chains that make up the complex. Similarly, human body can be visualized as connected rigid parts, much like a deformable part-based model [116]. The graph of the skeleton of human body can be divided into parts, where each subgraph represents a part of the human body.

In general, a part-based graph can be constructed as a combination of subgraphs where each subgraph has certain properties that define it. Let us consider that a graph $\mathcal{G}$ has been divided into n partitions. Formally:

$$\mathcal{G} = \bigcup_{p \in \{1,\dots,n\}} \mathcal{P}_p \mid \mathcal{P}_p = (\mathcal{V}_p, \mathcal{E}_p) \tag{5.5}$$

$\mathcal{P}_p$ is the partition (or subgraph) p of the graph $\mathcal{G}$. We consider scenarios in which the partitions can share vertices or have edges connecting them. We proceed to explain how the part-based graph convolution is defined for the part-based graph.

5.3.2 Part-based Graph Convolutions

In essence, graph convolutions over parts are aimed at capturing high-level properties of parts and learn the relations between them. In a Deformable Part-based Model, different parts are identified and relations between them are learned through the deformation of the connections between them. Similarly, graph convolutions over a part identifies the properties of that subgraph and an aggregation across subgraphs learns the relations between them. For a part-based graph, convolutions for each part are performed separately and the results are combined using an aggregation function $\mathcal{F}_{agg}$. Using $\mathcal{F}_{agg}$ over edges across partitions:

$$\mathbf{Y}_p(v_i) = \sum_{v_j \in \mathcal{N}_{kp}(v_i)} \mathbf{W}_p(\mathbf{L}_p(v_j))\mathbf{X}_p(v_j),\ p \in \{1, \ldots, n\} \tag{5.6}$$

$$\mathbf{Y}(v_i) = \mathcal{F}_{agg}(\mathbf{Y}_{p1}(v_i), \mathbf{Y}_{p2}(v_j)) \mid (v_i, v_j) \in \mathcal{E}_{(p1,p2)},\ (p1, p2) \in \{1, \ldots, n\} \times \{1, \ldots, n\} \tag{5.7}$$

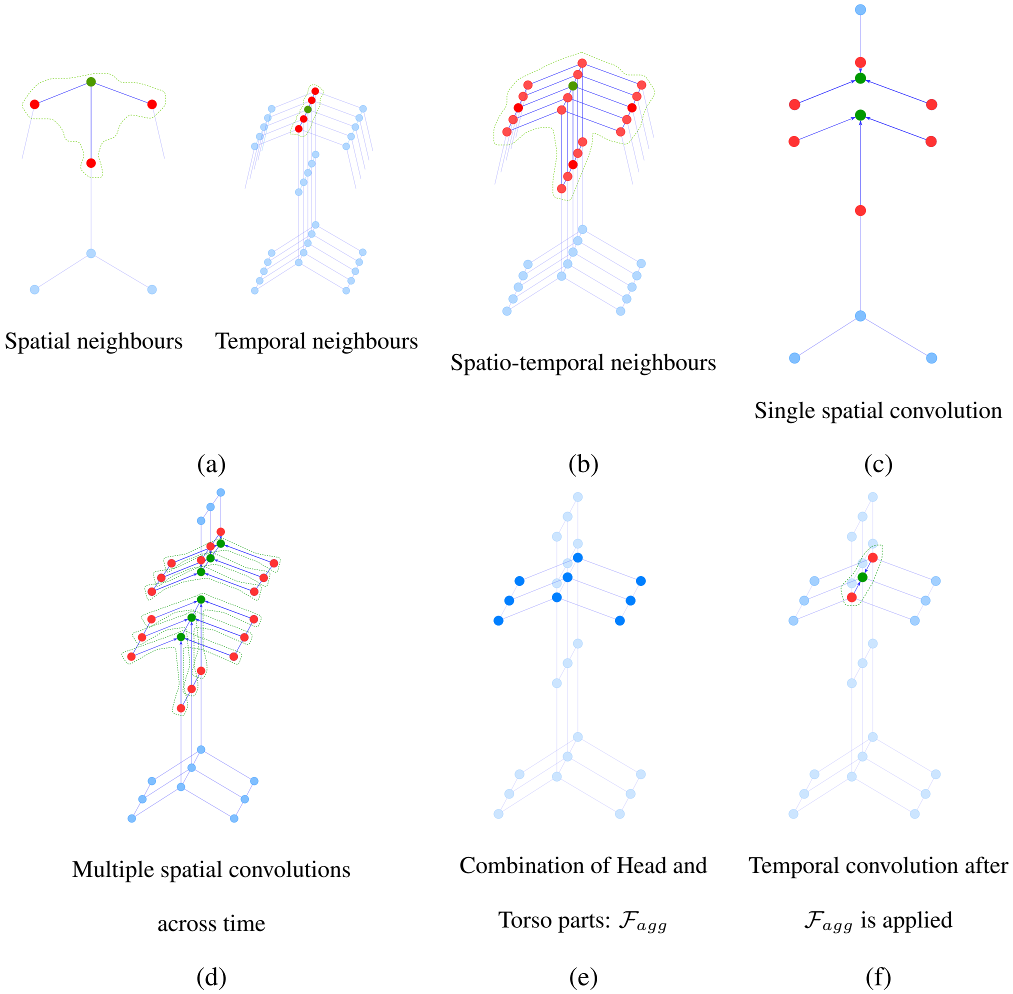
Using $\mathcal{F}_{agg}$ for common vertices across partitions:

$$\mathbf{Y}(v_i) = \mathcal{F}_{agg}(\mathbf{Y}_{p1}(v_i), \mathbf{Y}_{p2}(v_i)) \mid (p1, p2) \in \{1, \ldots, n\} \times \{1, \ldots, n\} \tag{5.8}$$

The convolution parameters $\mathbf{W}_p$ can be shared across parts or kept separate, while the neighbors of v_i only in that part $(\mathcal{N}_{kp}(v_i))$ are considered. In order to combine the information across parts, the function $\mathcal{F}_{agg}$ combines information at shared vertices (equation 5.8) or shares information through edges crossing parts (equation 5.7, $\mathcal{E}_{(p1,p2)}$ contains all edges connecting parts p1 and p2), according to the partition configuration. A sophisticated $\mathcal{F}_{agg}$ can be employed to make the model powerful. Using graph convolutions, part-based graph models can learn rich representations and we demonstrate the strength of this model through application to action recognition from S-videos.

5.4 Spatio-temporal Part-based Graph Convolutions

The S-videos are represented as spatio-temporal graphs. In order to include the temporal dimension, corresponding joints in each part are connected temporally. Figure 5.3(b) shows the spatio-temporal graph for *torso* over *five* frames. Adapting select-assemble-normalize (PATCHY-SAN) proposed by Niepert *et al.* [8] we present an overview of convolution formulation for our spatio-temporal graph by extending ideas from section 5.3.2. For in-depth understanding, we refer the reader to [8]. We perform a spatial convolution on each partition following equation 5.6, combine the convolved partitions using $\mathcal{F}_{agg}$ and perform temporal convolution on the graph obtained by aggregating the partitions. In effect, we spatially convolve each partition independently for each frame, aggregate them at each frame and perform temporal convolution on the temporal dimension of the aggregated graph. For a possible partitioning of human skeleton, this phenomenon is shown in Figure 5.3(c) for spatial convolution for a vertex common to torso and head, 5.3(d) for spatial convolutions in different frames, 5.3(e) for applying $\mathcal{F}_{agg}$ on head + torso and 5.3(f) for convolution on temporal dimension of the combined graph.



Spatial neighbours Temporal neighbours

(a)

Spatio-temporal neighbours

(b)

Single spatial convolution

(c)

Multiple spatial convolutions across time

(d)

Combination of Head and Torso parts: $\mathcal{F}_{agg}$

(e)

Temporal convolution after $\mathcal{F}_{agg}$ is applied

(f)

Figure 5.3: Spatio-temporal neighborhood for root node (in green) and depiction of convolutions in space and time dimensions. Effect of application of $\mathcal{F}_{agg}$ is shown, where the common vertices are in darker shade.

We first define the spatial and temporal neighborhood of a vertex in spatio-temporal graph and assign labels to the vertices in the neighborhoods, which is required to perform convolutions. For each vertex, we use 1-neighborhood $(k = 1)$ for spatial dimension $(\mathcal{N}_1)$ as the skeleton graph is not very large and a τ-neighborhood $(k = \tau)$ for the temporal dimension $(\mathcal{N}_\tau)$. Figure 5.3(a) (dashed polygons) shows the spatial & temporal neighborhood for a **root** vertex. The different neighborhood sets for our model are defined as $(\mathbf{d}(v_i, v_j)$ = length of shortest path between v_i and v_j):

$$\mathcal{N}_{1p}(v_i) = \{v_j \mid \mathbf{d}(v_i, v_j) \ \leq \ 1, \ v_i, v_j \in \mathcal{V}_p\} \tag{5.9}$$

$$\mathcal{N}_\tau(v_{it_a}) = \{v_{it_b} \mid \mathbf{d}(v_{it_a}, v_{it_b}) \ \leq \ \left\lfloor \frac{\tau}{2} \right\rfloor\} \tag{5.10}$$

where, t_a & t_b represent two time instants and $p \in \{1, \ldots, n\}$ is the partition index. The set of vertices $\mathcal{V}_p$ differs for each part, with some vertices shared between parts (Figure 5.1(c)). As temporal convolution is performed on the aggregated spatio-temporal graph, $\mathcal{N}_\tau$ is not part-specific. Figure 5.3(a) shows the spatial and temporal neighborhoods for a **root** vertex in *torso*. For ordering vertices in the receptive fields (or neighborhoods), we use a single label spatially ($\mathbf{L}_S : \mathcal{V} \rightarrow \{0\}$) to weigh vertices in $\mathcal{N}_{1p}$ of each vertex equally and τ labels temporally ($\mathbf{L}_T : \mathcal{V} \rightarrow \{0, \ldots, \tau - 1\}$) to weigh vertices across frames in $\mathcal{N}_\tau$ differently. The labeling functions are defined as:

$$\mathbf{L}_S(v_{jt}) = \{0 \mid v_{jt} \in \mathcal{N}_{1p}(v_{it})\} \tag{5.11}$$

$$\mathbf{L}_T(v_{it_b}) = \{((t_b - t_a) + \left\lfloor \frac{\tau}{2} \right\rfloor) \mid v_{it_b} \in \mathcal{N}_\tau(v_{it_a})\} \tag{5.12}$$

Using the labeled spatial and temporal receptive fields, we define the spatial and temporal convolutions as (adapted from [77]):

$$\mathbf{Y}_p(v_{it}) = \sum_{v_{jt} \in \mathcal{N}_{1p}(v_{it})} \mathbf{A}_p(i, j)\, \mathbf{Z}_p(v_{jt}) \mid p \in \{1, \ldots, 4\} \tag{5.13}$$

$$\mathbf{Z}_p(v_{jt}) = \mathbf{W}_p(\mathbf{L}_S(v_{jt}))\, \mathbf{X}_p(v_{jt}) \tag{5.14}$$

$$\mathbf{Y}_S(v_{it}) = \mathcal{F}_{agg}(\{\mathbf{Y}_1(v_{it}), \ldots, \mathbf{Y}_n(v_{it})\}) \tag{5.15}$$

$$\mathbf{Y}_T(v_{it_a}) = \sum_{v_{jt_b} \in \mathcal{N}_\tau(v_{it_a})} \mathbf{W}_T(\mathbf{L}_T(v_{it_b}))\, \mathbf{Y}_S(v_{it_b}) \tag{5.16}$$

where, $\mathbf{A}_p$ is a normalized adjacency matrix as explained in section 5.3 for part p. $\mathbf{L}_S$ for each part is same but $\mathcal{N}_{1p}$ is part-specific. $\mathbf{W}_p \in \mathbb{R}^{C' \times C \times 1 \times 1}$ is a part-specific channel transform kernel (pointwise operation) and $\mathbf{W}_T \in \mathbb{R}^{C' \times C' \times \tau \times 1}$ is the temporal convolution kernel. $\mathbf{Z}_p$ is the output from applying $\mathbf{W}_p$ on input features $\mathbf{X}_p$ at each vertex. $\mathbf{Y}_S$ is the output obtained after aggregating all partition graphs at one frame and $\mathbf{Y}_T$ is the output after applying temporal convolution on $\mathbf{Y}_S$ output of τ frames. We use a weighted sum fusion as our $\mathcal{F}_{agg}$:

$$\mathcal{F}_{agg}(\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}) = \sum_i \mathbf{W}_{agg}(i)\, \mathbf{Y}_i \tag{5.17}$$

Human skeleton can be divided into two major components: (1) Axial skeleton and (2) Appendicular skeleton. The body parts included in these two components are shown in Figure 5.1(b). Human skeleton can be divided into parts based on these components. Different division schemes are shown in Figure 5.1(b), 5.1(c) and 5.1(d) and we use these schemes for experiments to test our PB-GCN.

For the final representation, we divide the human skeleton into *four* parts: **head**, **hands**, **torso** and **legs**, which corresponds to a division scheme where each of the axial and appendicular skeleton are divided into upper and lower components, as illustrated in Figure 5.1(c). We consider left and right parts of hands and legs together in order to be agnostic to *laterality* [117] (handedness / footedness) of the human when performing an action. To show how being agnostic to laterality is helpful, we divide the upper and lower components of appendicular skeleton into left and right (shown in Figure 5.1(d)), resulting in six parts and show results on it. To cover all natural connections between joints in skeleton graph, we include an overlap of atleast one joint between two adjacent parts. For example, in Figure

(a) Performance with number of parts

#Parts	Accuracy	
	CS	CV
One	79.4	87.9
Two	80.2	88.4
Four	**82.8**	**90.3**
Six	81.4	89.1

(b) Performance with various signals for best & worst number of parts

Signals	Accuracy			
	#Parts=1		#Parts=4	
	CS	CV	CS	CV
J_{loc}	79.4	87.9	82.8	90.3
$\mathbf{D}_R$	83.6	87.7	84.6	88.4
$\mathbf{D}_T$	84.3	91.6	85.4	92.6
$\mathbf{D}_R\|\|\mathbf{D}_T$	**85.6**	**91.8**	**87.5**	**93.2**

Table 5.1: Performance comparison for different number of parts in the skeleton graph and signals at vertices using our PB-GCN, on NTURGB+D [13] (CS: Cross Subject, CV: Cross View). The symbols for signals, J_{loc}: Absolute 3D joint locations, $\mathbf{D}_R$: Relative coordinates, $\mathbf{D}_T$: Temporal displacements and $\mathbf{D}_R||\mathbf{D}_T$: Concatenation of $\mathbf{D}_R$ and $\mathbf{D}_T$.

5.1(c), shoulder joints are common between the head and hands. For the lower appendicular skeleton (viz. legs), we also include the joint at the base of spine to get a good overlap with lower axial skeleton.

Architecture and Implementation We represent each subgraph by its adjacency matrix, normalized by corresponding degree matrix $\mathcal{D}$. Our model takes as input a tensor having features for each vertex in the spatio-temporal graph of S-video and outputs a vector of class scores for the video. The architecture of the graph convolutional network is similar to Yan *et al.* [9] and consists of 9 spatio-temporal graph convolution units (each unit with the *four* $\mathbf{W}_p$ kernels, *one* $\mathbf{W}_T$ kernel and a residual) with an initial spatio-temporal head unit, based on a Resnet-like model [118]. First three layers have 64 output channels, next three have 128 and last three have 256. We also use a learnable edge weight mask for learning edge weights in each subgraph [9]. We use the Pytorch framework [119] for our implementation. The code and models are made publicly available: `https://github.com/dracarys983/pb-gcn`.

5.5 Geometric & Kinematic Signals

Yan *et al.* [9] use the 3D coordinates of each joint directly as the signal at each graph node. Relative coordinates [59, 60] and temporal displacements [108] of joints have been used earlier for action recognition. Derived information like optical flow and Manhattan line map has been found useful on RGB images also [55, 107]. Even a CNN framework can be more effective and efficient if relevant derived information is supplied as input to the network.

We use a signal at each node that combines temporal displacements across time and relative coordinates, with respect to shoulders and hips [60]. This representation provides translation invariance to the representation [62] and improves skeletal action recognition performance significantly. Figure 5.1(a) illustrates the computation of the two signals for a single skeleton video frame. We show the effect of relative joint coordinates (geometric signal) and temporal displacements (kinematic signal) individually and the performance improvement obtained by using a combination of these signals for a baseline one-part model as well as our four part-based model in the Table 5.1(b). The improvement in performance obtained using the geometric and kinematic signals is noteworthy.

To calculate the relative coordinates, we use four reference joints: left shoulder, right shoulder, left hip and right hip. It has been shown that these joints are relatively stable and can therefore be used as reference joints for computing the relative coordinates. Consider the 3D location of these four joints as P_{ls}, P_{rs}, P_{lh} and P_{rh}, respectively. For a joint j with position P_j, the geometric feature vector is calculated as:

$$\begin{aligned}
R_{ls} &= P_j - P_{ls} \\
R_{rs} &= P_j - P_{rs} \\
R_{lh} &= P_j - P_{lh} \\
R_{rh} &= P_j - P_{rh} \\
R_j &= concat(R_{ls}, R_{rs}, R_{lh}, R_{rh})
\end{aligned} \tag{5.18}$$

where R_j is the final geometric feature vector that has the relative coordinates of the joint with respect to the four reference joints. We use relative coordinates in our work to instil an understanding of usefulness of geometric features. There are many different geometric quantities that have been used before. For instance, joint-line distances are used by Zhang *et al.* [59], which work better than relative coordinates according to their LSTM-based method. Hence, different variations may be tried and performance can be boosted even further.

To calculate the temporal displacements, the corresponding joints belonging to adjacent frames are subtracted. Consider the position of joint j at time instant t to be P_{jt} and the position at time instant $t+1$ to be $P_{j(t+1)}$. Note that, the increment in one time instant is considered as moving to the next frame in the skeletal video here. The kinematic feature vector for the joint is calculated as:

$$\mathbf{d}_{jt} = P_{j(t+1)} - P_{jt} \tag{5.19}$$

The size of geometric feature vector for a joint is four times the size of absolute 3D location feature, while the size of kinematic feature vector is the same as that.

In principle, the convolutional filters should learn weights such that this information is captured automatically. However, our experiments for action recognition using graph convolutional networks prove otherwise. We believe that a better design of convolutional architectures would overcome such a limitation, but in it's present incarnation, convolutional networks benefit from differential information and their performance is improved.

5.6 Experimental Setup and Results

We use SGD as the optimizer and run the training for 80 epochs (NTURGB+D) / 120 epochs (HDM05). We set the initial learning rate to 0.1 and all the experiments are run on a cluster with 4 Nvidia GTX 1080Ti GPUs. The batch size is set to 64. Learning rate decay schedule (set to decay by 0.1 at epochs 20, 50 and 70 for NTURGB+D, and at epoch 80 for HDM05) is finalized using a validation set. No augmentation is performed for any of the experiments, consistent with graph-based method [9]. We perform ablation studies on the large-scale NTURGB+D dataset (shown in Table 5.1) and then compare with state-of-the-art on both HDM05 and NTURGB+D using the best configuration of our model (shown in Table 5.2).

5.6.1 Datasets

NTURGB+D [13] This is currently the largest RGBD dataset for action recognition to the best of our knowledge. It has 56,880 video sequences shot with three Microsoft Kinect v2 cameras from different viewing angles. There are 60 classes among the action sequences and 3D coordinates of 25 joints are provided for each human skeleton tracked. There is a large variation in viewpoint, intra-class subjects and sequence lengths, which makes this dataset challenging. We remove 302 of the captured samples having missing or incomplete skeleton data. The protocol mentioned in Shahroudy *et al.* [13] is followed for comparisons with previous methods.

HDM05 [45] This dataset was captured by using an optical marker-based Vicon system. It contains 2337 action sequences ranging across 130 motion classes performed by *five* actors. This dataset currently has the largest number of motion classes. The actors are named “bd”, “bk”, “dg”, “mm” and “tr”, and 31 joints are annotated for each skeleton. This dataset is challenging due to intra-class variations induced by multiple realizations of same action and large number of motion classes. We follow the protocol given in [14] which is used by recent deep learning methods.

5.6.2 Discussion

PART-BASED GRAPH MODEL: Our motivation to use a part-based graph model is derived primarily from the fact that human actions are made up of “gestures” which represent motion of a body part. The seminal success of DPMs [116] in detecting humans in images reinforces the motivation further. We discuss the effect of proposed spatio-temporal part-based graph model below.

(a) How many parts to have? We start with a coarse-grained scheme where entire skeleton is a single part and progress towards finer representations. The different partitions are, *two parts*: dividing skeleton into axial and appendicular skeleton, *four parts*: as explained in section 5.4 and *six parts*: Assigning left and right in hands and legs. The feature at each vertex in the input is 3D coordinate of the corresponding joint. From Table 5.1(a), we can see that using two parts improves over one and four improves over two. This shows that partitioning the skeleton graph into subgraphs with useful properties helps. However,

(a) NTURGB+D

Methods	Accuracy	
	CS	CV
ST Attention [58]	73.4	81.2
GCA-LSTM [5]	74.4	82.8
TCN [61]	74.3	83.1
VA-LSTM [120]	79.4	87.6
CNN + MTLN [60]	79.6	84.8
Deep STGC [64]	74.9	86.3
STGCN [9]	81.5	88.3
PB-GCN	**87.5**	**93.2**

(b) HDM05

Methods	Accuracy
SPDNet [14]	61.45 ± 1.12
Lie Group [41]	70.26 ± 2.89
LieNet [76]	75.78 ± 2.26
P-LSTM [13]	73.42 ± 2.05
Deep STGC [64]	85.29 ± 1.33
STGCN [9]	82.13 ± 2.39
PB-GCN	**88.17 ± 0.99**

Table 5.2: Performance comparison with previous methods on two benchmark datasets. The top group of results correspond to non-graph based methods and the middle corresponds to GCN based methods. PB-GCN is our part-based graph convolutional network. Evaluation protocols used: CS (Cross Subject) and CV (Cross View) for NTURGB+D [13]; 10-fold cross sample validation for HDM05 [14].

dividing upper and lower skeletons into left and right in four part scheme does not improve performance, as per our intuition about *laterality* mentioned in section 5.4. This experiment suggests that part-based model improves performance over single part and being agnostic to laterality is helpful. Our final model uses the *four* part division of the human skeleton.

(b) Comparison to graph-based models From Table 5.2(a) and Table 5.1(b), it can be seen that our part-based model performs better than graph based model of Yan *et al.* [9] even when using J_{loc} as the feature at each vertex. The graph construction in [9] uses a spatial partitioning scheme for their final model which divides the skeleton graph egde set into several partitions, while the vertex set has no partitions and contains all the joints. The difference in our model is that we divide the *entire* skeleton into *smaller* parts similar to human body parts and hence we use different edge set and vertex set for each part. Compared to graph based model of Li *et al.* [64], our model performs significantly better on NTURGB+D as well as HDM05. However, it is possible that this is because the number of layers in the network in [64] is much smaller (2 vs 9) compared to our model. Our model outperforms both the previous graph based models proposed for skeleton action recognition on the two datasets.

GEOMETRIC + KINEMATIC SIGNALS: Providing an explicit cue to a convolutional network, such as optical flow when performing action recognition from RGB videos [53], which is significant for the task at hand helps learn a richer representation by focusing on the cue. This motivates the use of geometric and kinematic features for skeletal action recognition. For the final configuration of our model, we concatenate the geometric and kinematic signals.

(a) Kinematic: temporal displacements Temporal displacements provide information about the amount of motion happening between two frames. This information is synonymous to 3D scene flow of a very sparse set of points. We hypothesize that these displacements provide explicit motion information (like optical flow) which makes the model consider displacements as strong features and learn from them. Improvement in performance using this signal can be seen from Table 5.1(b), for both four-part as well as one-part model across both splits of NTURGB+D.

(b) Geometric: relative coordinates These provide translation invariant features as explained in [62] and they have been used effectively to encode skeletons by Ke *et al.* [60] into images. Also, Zhang *et al.* [59] used relative coordinates as a geometric feature which performs much better than 3D joint locations using a simple stacked LSTM network. We can see improvements in performance provided by relative coordinates in Table 5.1(b) for both global (one part) and four part-based models, which are the worst and best performing models according to Table 5.1(a).

5.6.3 Comparison to state of the art

NTURGB+D: For this dataset, we outperform all previous state-of-the-art methods by a large margin. Even without using the signals introduced in section 5.5, we outperform the previous methods which can be seen in Table 5.1(b) (J_{loc} results). We outperform the previous state-of-the-art graph based method of Yan *et al.* [9] (STGCN) which is also the state-of-the-art for skeleton based action recognition to the best of our knowledge, by a margin of ˜6% and ˜5% for the two protocols.

HDM05: This is a ˜20x smaller dataset compared to NTURGB+D but contains more than twice the number of classes in NTURGB+D. The length of sequences in this dataset is longer and some of the action classes have only one sequence [121]. Using the protocol of [14] is therefore very challenging, on which we obtain state-of-the-art results using our model. We outperform the previous state-of-the-art Deep STGC [64], which is a network based on spectral graph convolutions for skeleton action recognition by ˜3% at the mean accuracy.

5.7 Analysis of different models

In this document, we present findings from further quantitative analysis on the action recognition results. Specifically, we compute the confusion matrices of the performance of different models and explain the useful model properties based on our observations. We find that graph-based models can understand actions which involve more motion better than those where skeleton motion is very less and contains object interactions. We also show the importance of using geometric and kinematic features instead of 3D joint locations by performing an experiment on graph-based model of Yan *et al.* [9].

Model	Accuracy	
	CS	CV
Yan [9] (model-2)	81.5	88.3
Model-2 + $\mathbf{D}_R \| \| \mathbf{D}_T$	86.3	92.1

Table 5.3: Results on NTURGB+D for model-2 [9], with and without the combined signal $\mathbf{D}_R||\mathbf{D}_T$ (relative coordinates and temporal displacements).

5.7.1 Quantitative Analysis

We compute the confusion matrices for performance of our part-based graph model, graph model using only one part and Yan's graph model [9]. We did not include Li's graph model [64] as no code has been provided by the authors to reproduce the results. The performance for cross subject (CS) evaluation protocol is considered as it is more challenging than the cross view (CV) evaluation protocol. The confusion matrices for different models are shown in Figure 5.4 (model-1), 5.5 (model-2) and 5.6 (model-3). The recognition accuracy for each of these models for cross subject (CS) evaluations is 85.6, 87.5 and 81.5 respectively. The model corresponding to Figure 5.4 is a one-part graph model which does not divide the skeleton graph into parts and it takes a combination of relative joint coordinates $\mathbf{D}_R$ and temporal displacements $\mathbf{D}_T$ as input. The model corresponding to 5.5 is our four-part graph model with $\mathbf{D}_R$ and $\mathbf{D}_T$ as input. Finally, Figure 5.6 corresponds to graph-based model introduced in Yan *et al.* [9] for skeleton action recognition. We proceed to identifying the action classes for which the recognition performance is bad, explain what the reasons are for such performance, propose a possible solution and then compare performance across different classes for models with respect to model-2.

5.7.2 Commonly confused classes

The confusion matrices have boxes marked around certain values. These boxes represent the confused classes which are consistent across all models. For example, one of the boxes is around action classes 11 & 12, which correspond to "reading" and "writing" actions. These actions are mostly confused amongst each other and also with actions such as "playing with the phone / tablet" or "typing on a keyboard" (actions 29 & 30 present in the other marked box) which is clear from the confusion matrices. In all these actions, there is almost no skeleton motion and the differences are manifested in the form of interaction with different objects. Due to these properties, models using skeleton information for recognizing actions give lower performance for these action classes as they do not have access to object information. A possible approach to overcome this limitation on recognition potential is to use RGB information along with skeleton information in order to get information about objects as well.

5.7.3 Model-1 vs Model-2

Model-2 improves over Model-1 by using a part-based graph representation instead of considering the entire graph as one part. Model-2 achieves better recognition performance by improving over action classes such as "brushing teeth" (class 3), "cheer up" (class 22), "make a phone call/answer phone" (class 28), etc. These actions have a strong correlation with movement of both hands and legs. Due to this correlation, our part-based graph model is able to achieve better performance as it learns from these parts specifically and uses an intuitive way to divide the human body into parts. Being agnostic to parts in human skeleton helps in learning a global representation but learning importance of parts using such a model is difficult, compared to a part-based model.

5.7.4 Model-3 vs Model-2

Spatio-temporal model of Yan *et al.* [9] confuses the action of "clapping" as well along with the actions mentioned in section 5.7.2. The model proposed by Yan [9] partitions the edge set and uses the same vertex set for each partition of edge set. We believe that their model learns the importance of different edges in the skeleton graph and does not learn the importance of parts like our part-based graph model. In order to understand the influence of geometric and kinematic signals as input to a graph-based model, we use the signals on top of model-3 and we find that we get a boost in recognition performance for model-3. The recognition accuracy on NTURGB+D is shown in Table 5.3. This experiment shows that the signals help in improving recognition performance for different graph-models for skeleton action recognition.

5.7.5 Observations from analysis

Using a part-based model works better than using a model that does not partition the skeleton graph. However, using only skeletal data for action recognition is not enough as different actions might have similar dynamics of parts in the skeleton but different object interactions. In such cases, RGB information can be used to disambiguate interactions with objects. Providing the network with a cue that is suited to work well for the task at hand, viz. relative coordinates and temporal displacements for skeletal action recognition, can improve recognition performance by a large amount as we show in our experiment on previous state-of-the-art model for NTURGB+D [9].

5.8 Conclusion

In this work, we define a partition of skeleton graph on which spatio-temporal convolutions are formalized through a part-based GCN for the task of action recognition. Such a part-based GCN learns the relations between parts and understands the importance of each part in human actions more effectively than a model that considers entire body as a single graph. We also demonstrate the benefit of giving

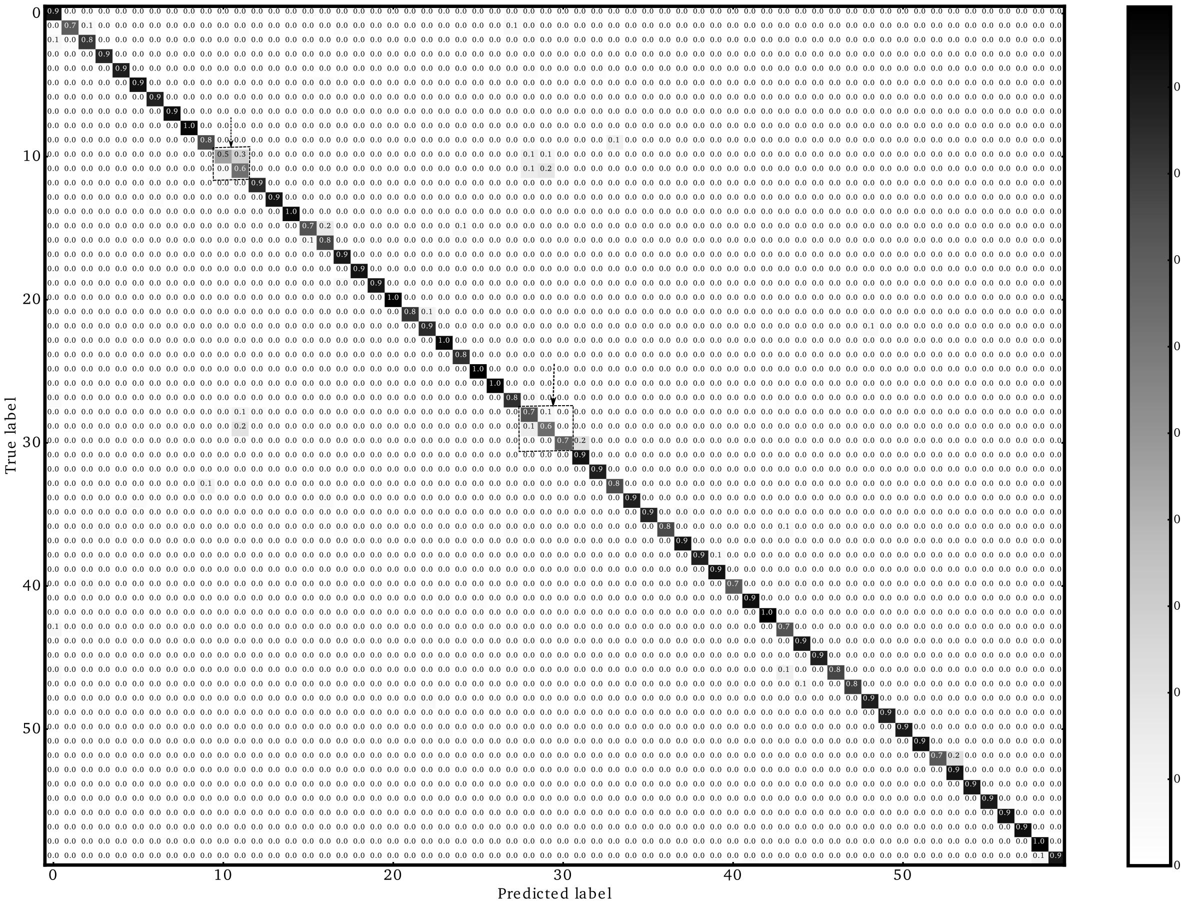


Figure 5.4: Confusion matrix for model with one part and combined geometric + kinematic features as input.

explicit cues to the convolutional model which are significant from the point of view of the task at hand, such as relative coordinates and temporal displacements for skeletal action recognition. As a result, our model achieves state-of-the-art performance on two challenging action recognition datasets. As a future work, we would like to explore the use of part-based graph model for tasks other than action recognition, such as object detection, measuring image similarity, etc.

5.9 Applications of Graph Convolution Network: Head Pose Estimation

Monocular head pose estimation requires learning a model that computes the intrinsic Euler angles for pose (yaw, pitch, roll) from an input image of human face. Annotating ground truth head pose angles for images in the wild is difficult and requires ad-hoc fitting procedures (which provides only coarse and approximate annotations). This highlights the need for approaches which can train on data captured in controlled environment and generalize on the images in the wild (with varying appearance and illumination of the face). Most present day deep learning approaches which learn a regression function directly on the input images fail to do so. To this end, we propose to use a higher level representation to regress the head pose while using deep learning architectures. We extract the 2D locations of five facial keypoints, namely left ear, right ear, left eye, right eye and nose, and estimate the head-pose using a Graph Convolutional Network (GCN). The input to the GCN are the exact locations of the localized facial keypoints.

We show the graph structure of the facial keypoints and some results supporting our claim about generalizing to images in the wild in Figure 5.7 and Figure 5.8. This shows the ability of geometric deep learning frameworks in learning powerful spatial relationships from data lying in non-Euclidean space through graph convolutional networks.

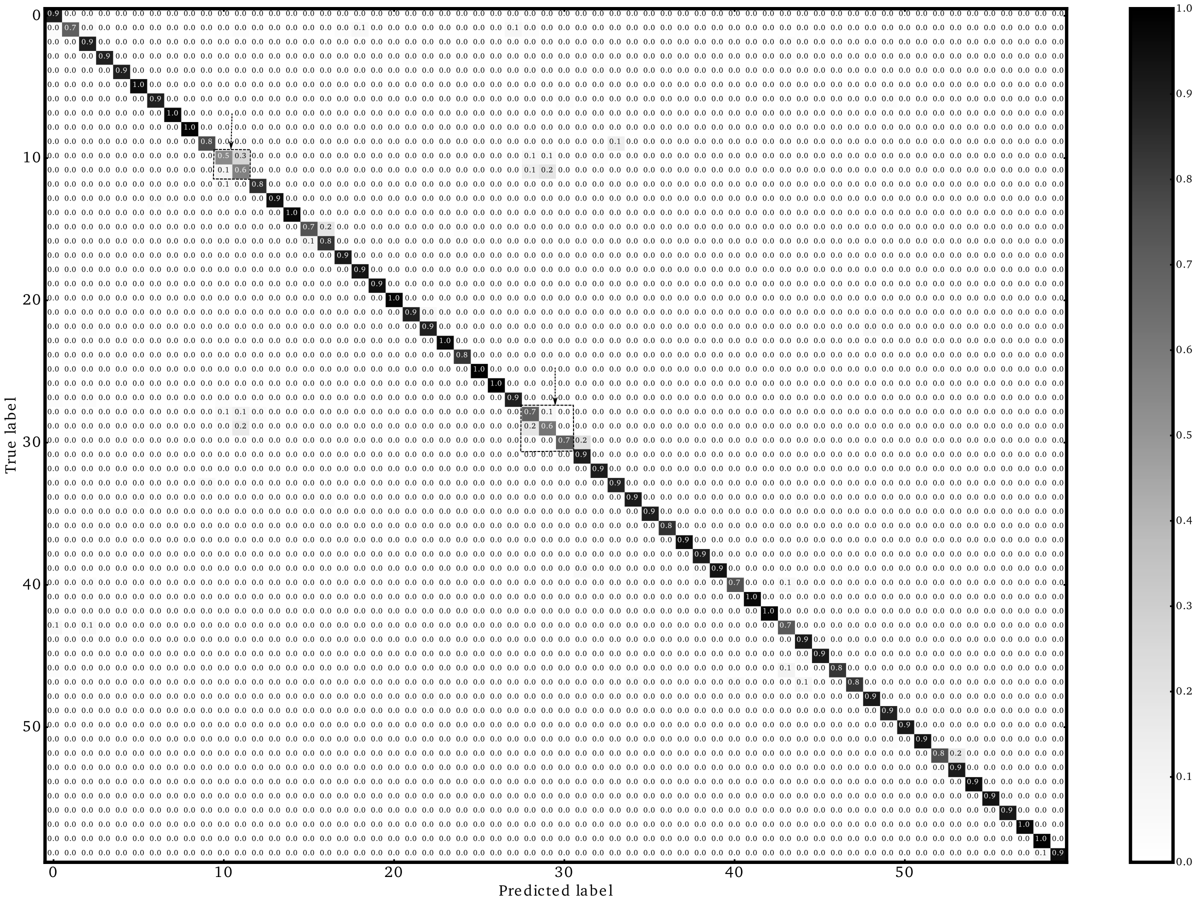


Figure 5.5: Confusion matrix for model with four parts and combined geometric + kinematic features as input.

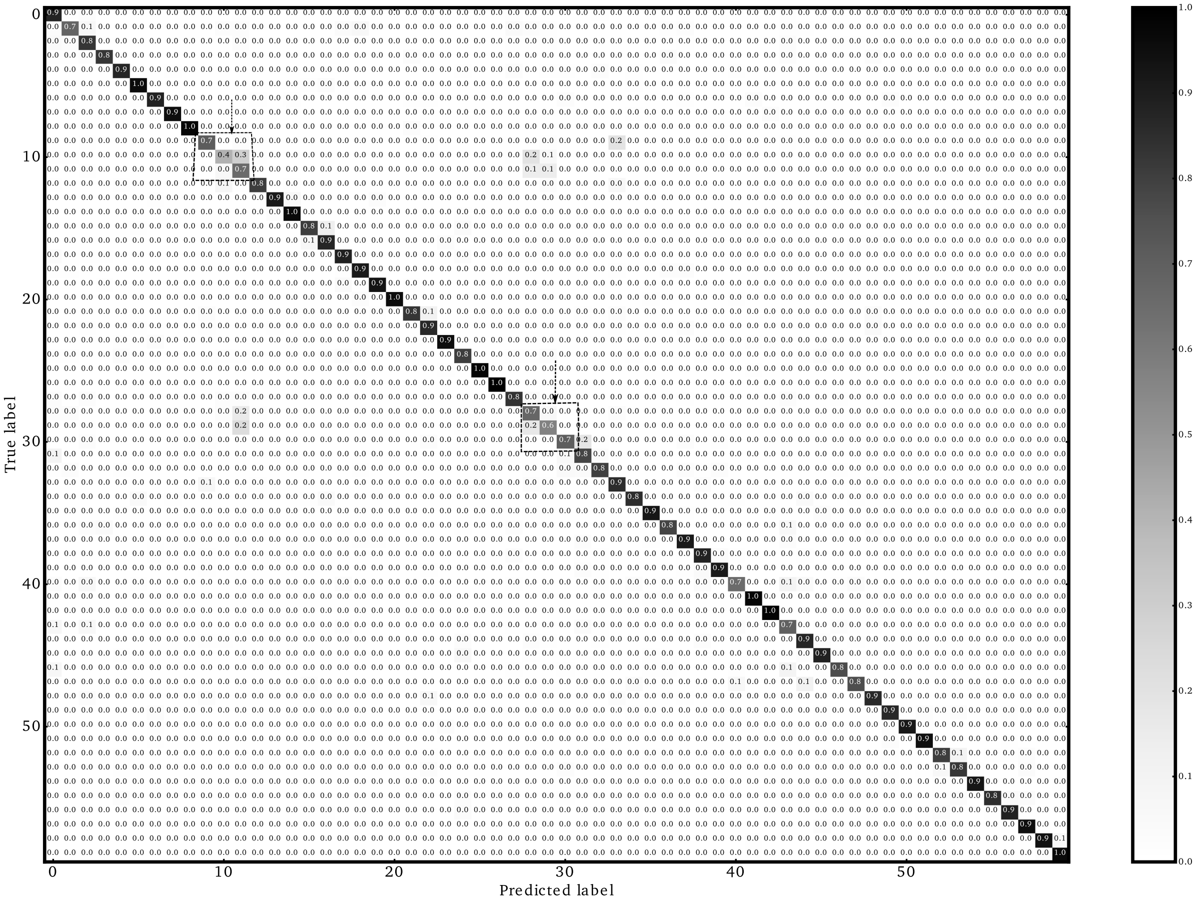
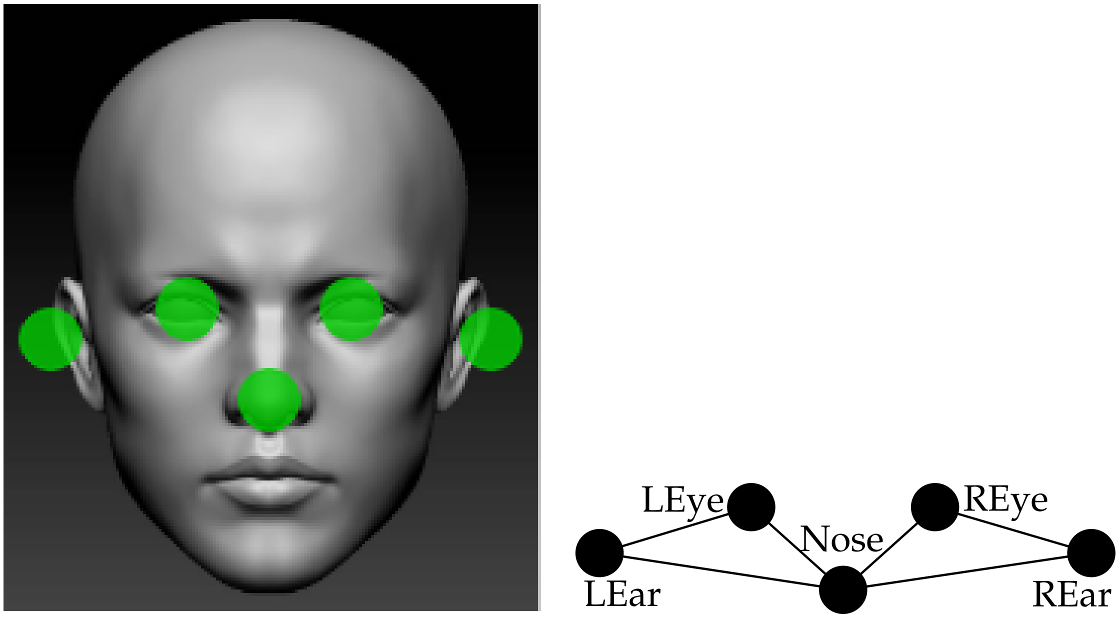


Figure 5.6: Confusion matrix for Yan's graph-based model [9] having 3D joint locations as input signals.



(a) (b)

Figure 5.7: (a) The facial keypoints selected to regress the head pose using our approach, (b) The graph structure formed from the chosen set of facial keypoints for graph convolutional network. (LEye = left eye, REye = right eye. Same follows for ears).

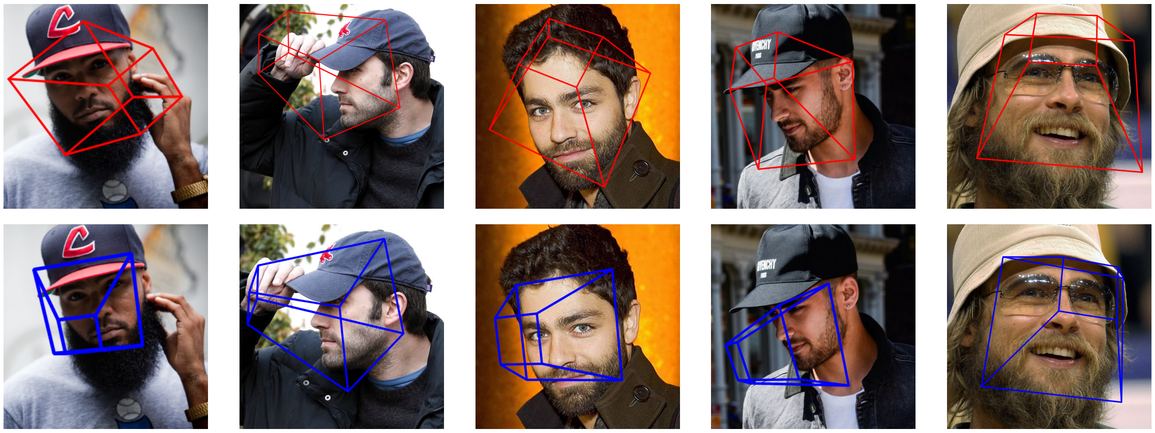


Figure 5.8: Estimation of head pose using three different models (all trained on BIWI), on unseen images taken from the web. **Top row**: Results for CNN-based model [10] which takes RGB images as input, **Bottom row**: Results for GCN-based framework which takes locations of keypoints as input. It can be seen that our GCN-based model generalizes much better than a CNN-based model directly trained on face images.

Chapter 6

Conclusion and Future Work

In this thesis, we explain the development of a framework for spatio-temporal representation of actions from skeletal videos of humans providing information about human pose in 3D through locations of a fixed number of skeleton joints. The foundations of our framework lie on the physical realities of human actions, which are initially captured using angular momentum of pose changes taking place during performance of actions and further computing a compact representation based on relative joints and angular momentum. As such an approach for the realization of the framework based on geometric and kinematic signals is incapable of scaling with availability of more data, we explore the use of a deep learning method: graph convolutional networks (GCNs).

This geometric deep learning approach capitalizes on the structural information contained in the human skeleton represented as a sparse spatial graph through spatio-temporal graph convolutions. We propose a novel part-based graph convolutional network (PB-GCN) which divides the human skeletal graph into meaningful parts and learns action representations using a GCN. In addition to that, the former conclusions regarding geometric and kinematic features inspire the use of relative coordinates and temporal displacements (angular momentum is a higher level temporal information) instead of using 3D joint locations as features at the vertices in the skeletal graph. Use of such signals prove to be beneficial to the performance of recognition and boost the resultant accuracies for action recognition.

Our work points out an important property of deep learning methods: they are not as smart as we think they are. Providing explicit cues to deep learning frameworks that are helpful for the task at hand boost the performance of the framework. In the case of action recognition using human skeletal pose, geometric signals such as relative coordinates, that encode higher level spatial relationships and kinematic signals such as temporal displacements (higher order information like velocity and acceleration can also be used), that encode temporal evolution of spatial pose information, are helpful and boost performance of action recognition. The approaches proposed in this work also highlight one more aspect about recognizing actions: objects present in the scene are important as there are action involving interactions of subject with the objects. This cannot be captured using skeletal videos but using RGB information along with skeletal information can improve recognition accuracies by a good margin. Approaches for action recognition from skeletal videos capture spatial and temporal relationships from 3D skeletal data very effectively and combining them with cues from RGB data would assist in capturing human-object interactions. Hence, a joint model has the capability to perform better action recognition and is left to future works.

We also show results obtained through a GCN-based framework for head pose estimation, which provides better generalization properties than methods trained directly on RGB images for regressing the head pose. The input provided to our framework is an abstraction over the image (location of five facial keypoints) and it provides illumination as well as appearance invariance to estimation of head pose.

Related Publications

- **Kalpit Thakkar** and P J Narayanan, "*Segmentation and Representation of Skeletal Videos using Angular Momentum*", arXiV:preprint.
- **Kalpit Thakkar** and P J Narayanan, "*Part-based Graph Convolutional Networks for Action Recognition*", 29th British Machine Vision Conference, BMVC 2018, Newcastle, UK

Other Publications

- **Kalpit Thakkar***, **Aryaman Gupta***, Vineet Gandhi and P J Narayanan, "*Nose, Eyes and Ears: Head Pose Estimation by Locating Facial Keypoints*", Submitted to 14th Asian Conference on Computer Vision, ACCV 2018, Perth, Western Australia

  * Equal contribution

Bibliography

[1] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 3, pp. 257–267, 2001. xii, 13, 14

[2] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 12, pp. 2247–2253, 2007. xii, 13, 14

[3] A. Yilmaz and M. Shah, "Actions sketch: A novel action representation," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 984–989, IEEE, 2005. xii, 14

[4] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1110–1118, 2015. xiii, 16, 18

[5] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention lstm networks for 3d action recognition," in *CVPR*, 2017. xiii, 16, 18, 44, 46, 55

[6] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," in *European Conference on Computer Vision*, pp. 816–833, Springer, 2016. xiii, 16, 18, 44, 46

[7] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009. xiii, 22, 23

[8] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *International conference on machine learning*, pp. 2014–2023, 2016. xiii, 9, 22, 23, 24, 25, 26, 27, 44, 48, 49

[9] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," *AAAI Conference on Artificial Intelligence*, 2018. xiv, xv, 18, 44, 45, 47, 52, 54, 55, 56, 57, 58, 61

[10] H. Yang, W. Mou, Y. Zhang, I. Patras, H. Gunes, and P. Robinson, "Face alignment assisted by head pose estimation," in *BMVC*, 2015. xiv, 62

[11] 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*, pp. 9–14, June 2010. xv, 10, 30, 38, 39, 40, 42

[12] L. Xia, C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pp. 20–27, IEEE, 2012. xv, 10, 30, 38, 42, 43, 46

[13] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+d: A large scale dataset for 3d human activity analysis," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. xv, 9, 10, 16, 19, 30, 44, 52, 54, 55

[14] Z. Huang and L. J. Van Gool, "A riemannian network for spd matrix learning.," in *AAAI*, vol. 2, p. 6, 2017. xv, 22, 54, 55, 56

[15] N. Huggett, "Zeno's paradoxes," in *The Stanford Encyclopedia of Philosophy* (E. N. Zalta, ed.), Metaphysics Research Lab, Stanford University, summer 2018 ed., 2018. 1

[16] B. Lüderitz, "Etienne jules marey (1830–1904)," *Journal of Interventional Cardiac Electrophysiology*, vol. 12, no. 1, pp. 91–92, 2005. 1

[17] G. Hendricks, *Eadweard Muybridge: the father of the motion picture*. Secker & Warburg, 1975. 1

[18] E. Muybridge, *Animals in motion*. Courier Corporation, 2012. 1

[19] E. Muybridge, *The human figure in motion*. Courier Corporation, 2012. 1

[20] E. Muybridge, *Animal Locomotion: An Electro-photographic Investigation of Consecutive Phases of Animal Movements. 1872-1885*. UPenn Archives, 1887. 1

[21] E. Muybridge, *The male and female figure in motion: 60 classic photographic sequences*. Courier Corporation, 1984. 1

[22] G. Johansson, "Visual motion perception," *Scientific American*, vol. 232, no. 6, pp. 76–89, 1975. 1, 5, 11, 12

[23] N. H. Goddard, "The perception of articulated motion: recognizing moving light displays," tech. rep., ROCHESTER UNIV NY DEPT OF COMPUTER SCIENCE, 1992. 1

[24] C. Cedras and M. Shah, "A survey of motion analysis from moving light displays," in *CVPR*, pp. 214–221, Seattle, 1994. 1, 5, 13

[25] J. E. Boyd and J. J. Little, "Global versus structured interpretation of motion: moving light displays," in *nam*, p. 0018, IEEE, 1997. 1

[26] M. Pantic, A. Pentland, A. Nijholt, and T. S. Huang, "Human computing and machine understanding of human behavior: A survey," in *Artifical Intelligence for Human Computing*, (Berlin, Heidelberg), pp. 47–71, Springer Berlin Heidelberg, 2007. 2

[27] A. A. Chaaraoui, P. Climent-Pérez, and F. Flórez-Revuelta, "A review on vision techniques applied to human behaviour analysis for ambient-assisted living," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10873–10888, 2012. 2, 4

[28] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis, *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *Computer vision and pattern recognition (CVPR), 2011 IEEE conference on*, pp. 3153–3160, IEEE, 2011. 2

[29] S. Vishwakarma and A. Agrawal, "A survey on activity recognition and behavior understanding in video surveillance," *The Visual Computer*, vol. 29, no. 10, pp. 983–1009, 2013. 2

[30] A. Jaimes and N. Sebe, "Multimodal human–computer interaction: A survey," *Computer vision and image understanding*, vol. 108, no. 1-2, pp. 116–134, 2007. 2

[31] M. A. Goodrich, A. C. Schultz, *et al.*, "Human–robot interaction: a survey," *Foundations and Trends® in Human–Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2008. 2

[32] G. Hee Lee, F. Faundorfer, and M. Pollefeys, "Motion estimation for self-driving cars with a generalized camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2746–2753, 2013. 2

[33] S. Pillai and J. J. Leonard, "Towards visual ego-motion learning in robots," *arXiv preprint arXiv:1705.10279*, 2017. 2

[34] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer vision and image understanding*, vol. 104, no. 2-3, pp. 90–126, 2006. 3, 4

[35] R. Poppe, "Vision-based human motion analysis: An overview," *Computer vision and image understanding*, vol. 108, no. 1-2, pp. 4–18, 2007. 3, 4

[36] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *IEEE Transactions on Circuits and Systems for Video technology*, vol. 18, no. 11, p. 1473, 2008. 3

[37] X. Wang, A. Farhadi, and A. Gupta, "Actions˜ transformations," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2658–2667, 2016. 4

[38] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," *Image and vision computing*, vol. 60, pp. 4–21, 2017. 4, 5, 14, 16

[39] P. Wang, W. Li, P. Ogunbona, J. Wan, and S. Escalera, "Rgb-d-based human motion recognition with deep learning: A survey," *Computer Vision and Image Understanding*, 2018. 6

[40] J. Liu, N. Akhtar, and A. Mian, "Skepxels: Spatio-temporal image representation of human skeleton joints for action recognition," *arXiv preprint arXiv:1711.05941*, 2017. 8

[41] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 588–595, 2014. 9, 22, 46, 55

[42] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. D. Bimbo, "3-d human action recognition by shape analysis of motion trajectories on riemannian manifold," *IEEE Transactions on Cybernetics*, vol. 45, pp. 1340–1352, July 2015. 9, 22

[43] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013. 9

[44] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems*, pp. 3844–3852, 2016. 9, 28, 44, 46, 48

[45] M. Mller, T. Rder, M. Clausen, B. Eberhardt, B. Krger, and A. Weber, "Documentation mocap database hdm05," 2007. 9, 10, 54

[46] C. D. Barclay, J. E. Cutting, and L. T. Kozlowski, "Temporal and spatial factors in gait perception that influence gender recognition," *Perception & psychophysics*, vol. 23, no. 2, pp. 145–152, 1978. 11

[47] G. Johansson, "Spatio-temporal differentiation and integration in visual motion perception," *Psychological research*, vol. 38, no. 4, pp. 379–393, 1976. 11

[48] P. Matikainen, M. Hebert, and R. Sukthankar, "Trajectons: Action recognition through the motion analysis of tracked features," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pp. 514–521, IEEE, 2009. 15

[49] I. Laptev, "On space-time interest points," *International journal of computer vision*, vol. 64, no. 2-3, pp. 107–123, 2005. 15

[50] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008. 15

[51] N. Dalal, B. Triggs, and C. Schmid, “Human detection using oriented histograms of flow and appearance,” in *European conference on computer vision*, pp. 428–441, Springer, 2006. 15

[52] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, “Action recognition by dense trajectories,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 3169–3176, IEEE, 2011. 15, 17

[53] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 568–576, Curran Associates, Inc., 2014. 15, 55

[54] C. Feichtenhofer, A. Pinz, and R. Wildes, “Spatiotemporal residual networks for video action recognition,” in *Advances in neural information processing systems*, pp. 3468–3476, 2016. 15

[55] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, “Temporal segment networks: Towards good practices for deep action recognition,” in *European Conference on Computer Vision*, pp. 20–36, Springer, 2016. 15, 45, 52

[56] M. Zanfir, M. Leordeanu, and C. Sminchisescu, “The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection,” in *2013 IEEE International Conference on Computer Vision*, pp. 2752–2759, Dec 2013. 15, 30, 42

[57] J. Shan and S. Akella, “3d human action segmentation and recognition using pose kinetic energy,” in *Advanced Robotics and its Social Impacts (ARSO), 2014 IEEE Workshop on*, pp. 69–75, IEEE, 2014. 16

[58] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, “An end-to-end spatio-temporal attention model for human action recognition from skeleton data.,” in *AAAI*, vol. 1, p. 7, 2017. 16, 44, 46, 55

[59] S. Zhang, X. Liu, and J. Xiao, “On geometric features for skeleton-based action recognition using multilayer lstm networks,” in *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pp. 148–157, IEEE, 2017. 16, 44, 45, 52, 53, 56

[60] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, “A new representation of skeleton sequences for 3d action recognition,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4570–4579, IEEE, 2017. 17, 44, 46, 52, 53, 55, 56

[61] T. S. Kim and A. Reiter, “Interpretable 3d human action analysis with temporal convolutional networks,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pp. 1623–1631, IEEE, 2017. 17, 55

[62] N. Verma, E. Boyer, and J. Verbeek, “Feastnet: Feature-steered graph convolutions for 3d shape analysis,” in *CVPR 2018-IEEE Conference on Computer Vision & Pattern Recognition*, 2018. 17, 53, 56

[63] S. Vashishth, S. S. Dasgupta, S. N. Ray, and P. Talukdar, “Dating documents using graph convolution networks,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1605–1615, Association for Computational Linguistics, 2018. 17

[64] C. Li, Z. Cui, W. Zheng, C. Xu, and J. Yang, “Spatio-temporal graph convolution for skeleton based action recognition,” *AAAI Conference on Artificial Intelligence*, 2018. 18, 44, 45, 47, 55, 56, 57

[65] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, “Convolutional networks on graphs for learning molecular fingerprints,” in *Advances in neural information processing systems*, pp. 2224–2232, 2015. 18

[66] E. P. Simoncelli and B. A. Olshausen, “Natural image statistics and neural representation,” *Annual review of neuroscience*, vol. 24, no. 1, pp. 1193–1216, 2001. 20

[67] D. J. Field, “What the statistics of natural images tell us about visual coding,” in *Human Vision, Visual Processing, and Digital Display*, vol. 1077, pp. 269–277, International Society for Optics and Photonics, 1989. 20

[68] P. Mehta and D. J. Schwab, “An exact mapping between the variational renormalization group and deep learning,” *arXiv preprint arXiv:1410.3831*, 2014. 20

[69] J. Bruna and S. Mallat, “Invariant scattering convolution networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013. 20

[70] M. Tygert, J. Bruna, S. Chintala, Y. LeCun, S. Piantino, and A. Szlam, “A mathematical motivation for complex-valued convolutional networks,” *Neural computation*, vol. 28, no. 5, pp. 815–825, 2016. 20

[71] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989. 20

[72] S. Mallat, “Group invariant scattering,” *Communications on Pure and Applied Mathematics*, vol. 65, no. 10, pp. 1331–1398, 2012. 21

[73] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, “Maxout networks,” *arXiv preprint arXiv:1302.4389*, 2013. 21

[74] D. Lazer, A. S. Pentland, L. Adamic, S. Aral, A. L. Barabasi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, *et al.*, “Life in the network: the coming age of computational social science,” *Science (New York, NY)*, vol. 323, no. 5915, p. 721, 2009. 21

[75] E. H. Davidson, J. P. Rast, P. Oliveri, A. Ransick, C. Calestani, C.-H. Yuh, T. Minokawa, G. Amore, V. Hinman, C. Arenas-Mena, *et al.*, “A genomic regulatory network for development,” *science*, vol. 295, no. 5560, pp. 1669–1678, 2002. 21

[76] Z. Huang, C. Wan, T. Probst, and L. Van Gool, “Deep learning on lie groups for skeleton-based action recognition,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6099–6108, IEEE computer Society, 2017. 22, 46, 55

[77] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016. 27, 28, 44, 46, 51

[78] L. Seidenari, V. Varano, S. Berretti, A. D. Bimbo, and P. Pala, “Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 479–485, June 2013. 30

[79] S. Baysal, M. C. Kurt, and P. Duygulu, “Recognizing human actions using key poses,” in *2010 20th International Conference on Pattern Recognition*, pp. 1727–1730, Aug 2010. 30

[80] F. Lv and R. Nevatia, “Single view human action recognition using key pose matching and viterbi path searching,” in *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*, pp. 1–8, IEEE, 2007. 30

[81] Y. Chen, Y. Deng, Y. Guo, W. Wang, Y. Zou, and K. Wang, “A temporal video segmentation and summary generation method based on shots’ abrupt and gradual transition boundary detecting,” in *2010 Second International Conference on Communication Software and Networks*, pp. 271–275, Feb 2010. 30

[82] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, “Summary transfer: Exemplar-based subset selection for video summarization,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 30

[83] J. Meng, H. Wang, J. Yuan, and Y.-P. Tan, “From keyframes to key objects: Video summarization by representative object proposal selection,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 30

[84] G. Boccignone, A. Chianese, V. Moscato, and A. Picariello, “Foveated shot detection for video segmentation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, pp. 365–377, March 2005. 30

[85] M. Meshry, M. E. Hussein, and M. Torki, “Linear-time online action detection from 3d skeletal data using bags of gesturelets,” in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–9, March 2016. 30, 40, 42

[86] B. X. Nie, C. Xiong, and S. C. Zhu, "Joint action recognition and pose estimation from video," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1293–1301, June 2015. 30

[87] L. Tao and R. Vidal, "Moving poselets: A discriminative and interpretable skeletal motion representation for action recognition," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pp. 303–311, Dec 2015. 30

[88] L. Wang, J. Zhang, L. Zhou, C. Tang, and W. Li, "Beyond covariance: Feature representation with nonlinear kernel matrices," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4570–4578, 2015. 30

[89] 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 *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pp. 1351–1357, AAAI Press, 2013. 30, 36, 42

[90] 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 *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pp. 2466–2472, AAAI Press, 2013. 30, 42

[91] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proceedings of the 18th International Conference on Pattern Recognition - Volume 03*, ICPR '06, (Washington, DC, USA), pp. 850–855, IEEE Computer Society, 2006. 33

[92] S. Zhang, X. Liu, and J. Xiao, "On geometric features for skeleton-based action recognition using multilayer lstm networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 148–157, March 2017. 36

[93] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3d action recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 36

[94] J. Liu, G. Wang, P. Hu, L. Y. Duan, and A. C. Kot, "Global context-aware attention lstm networks for 3d action recognition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3671–3680, July 2017. 36

[95] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1290–1297, June 2012. 42

[96] R. Slama, H. Wannous, and M. Daoudi, "Grassmannian representation of motion depth for 3d human gesture and action recognition," in *2014 22nd International Conference on Pattern Recognition*, pp. 3499–3504, Aug 2014. 42

[97] R. Slama, H. Wannous, M. Daoudi, and A. Srivastava, "Accurate 3d action recognition using learning on the grassmann manifold," *Pattern Recognition*, vol. 48, no. 2, pp. 556 – 567, 2015. 42, 43

[98] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 588–595, June 2014. 40, 42, 43

[99] O. Oreifej and Z. Liu, "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 716–723, June 2013. 42

[100] C. Ellis, S. Z. Masood, M. F. Tappen, J. J. Laviola, Jr., and R. Sukthankar, "Exploring the tradeoff between accuracy and observational latency in action recognition," *Int. J. Comput. Vision*, vol. 101, pp. 420–436, Feb. 2013. 42

[101] M. Jiang, J. Kong, G. Bebis, and H. Huo, "Informative joints based human action recognition using skeleton contexts," *Signal Processing: Image Communication*, vol. 33, no. Supplement C, pp. 29 – 40, 2015. 43

[102] A.-A. Liu, W.-Z. Nie, Y.-T. Su, L. Ma, T. Hao, and Z.-X. Yang, "Coupled hidden conditional random fields for rgb-d human action recognition," *Signal Processing*, vol. 112, no. Supplement C, pp. 74 – 82, 2015. Signal Processing and Learning Methods for 3D Semantic Analysis. 43

[103] M. Devanne, H. Wannous, P. Pala, S. Berretti, M. Daoudi, and A. D. Bimbo, "Combined shape analysis of human poses and motion units for action segmentation and recognition," in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, vol. 07, pp. 1–6, May 2015. 43

[104] C. Wang, Y. Wang, and A. L. Yuille, "Mining 3d key-pose-motifs for action recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2639–2647, June 2016. 43

[105] L. Chunhui, H. Yueyu, L. Yanghao, S. Sijie, and L. Jiaying, "Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding," *ACM Multimedia workshop*, 2017. 44

[106] C. Chen, R. Jafari, and N. Kehtarnavaz, "Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," in *IEEE International Conference on Image Processing (ICIP)*, 2015. 44

[107] C. Zou, A. Colburn, Q. Shan, and D. Hoiem, “Layoutnet: Reconstructing the 3d room layout from a single rgb image,” *arXiv preprint arXiv:1803.08999*, 2018. 45, 52

[108] M. Zanfir, M. Leordeanu, and C. Sminchisescu, “The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2752–2759, 2013. 45, 46, 52

[109] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, “Sequence of the most informative joints (smij): A new representation for human skeletal action recognition,” *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 24–38, 2014. 46

[110] L. Tao and R. Vidal, “Moving poselets: A discriminative and interpretable skeletal motion representation for action recognition,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 61–69, 2015. 46

[111] Y. Du, W. Wang, and L. Wang, “Hierarchical recurrent neural network for skeleton based action recognition,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1110–1118, 2015. 46

[112] Y. Du, Y. Fu, and L. Wang, “Skeleton based action recognition with convolutional neural network,” in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pp. 579–583, 2015. 46

[113] B. Li, M. He, Y. Dai, X. Cheng, and Y. Chen, “3d skeleton based action recognition by video-domain translation-scale invariant mapping and multi-scale dilated cnn,” *Multimedia Tools and Applications*, pp. 1–21, 2018. 46

[114] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo, “3-d human action recognition by shape analysis of motion trajectories on riemannian manifold,” *IEEE transactions on cybernetics*, vol. 45, no. 7, pp. 1340–1352, 2015. 46

[115] M. Li and H. Leung, “Graph-based approach for 3d human skeletal action recognition,” *Pattern Recognition Letters*, vol. 87, pp. 195–202, 2017. 47

[116] P. F. Felzenszwalb and D. P. Huttenlocher, “Pictorial structures for object recognition,” *Int. J. Comput. Vision*, vol. 61, no. 1, pp. 55–79, 2005. 48, 54

[117] Wikipedia, “Definition of laterality.” `https://en.wikipedia.org/wiki/Laterality`, 2015. 51

[118] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 52

[119] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017. 52

[120] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive recurrent neural networks for high performance human action recognition from skeleton data," *arXiv, no. Mar*, 2017. 55

[121] K. Cho and X. Chen, "Classifying and visualizing motion capture sequences using deep neural networks," *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 2, pp. 122–130, 2014. 56