Motion Analysis based on Significant Points

by

Nasim Hajari

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

University of Alberta

Abstract

Motion analysis is an important concept and it has extensive applications from surveillance, education, smart rooms, entertainment and so on. Different type of sensors can be used to capture the required motion information from different body parts. One can use eye-gaze system to capture and analyse eye motion or using Kinect or Leap Motion to capture 3d trajectories of important body keypoints. Even one can use a conventional digital camera for motion detection and action recognition purposes. Generally speaking motion data is dynamic data where the spatial features of the data change over time. These features can be the coordinates of significant points, such as fixation points of eyes, skeletal joints and so on, or the location of user defined features such as Histogram of Oriented Gradient (HOG), centre of Hough circles, centre of mass and so on. One can also analyse motion data in 1D, 2D or 3D space, depending on acquisition devices. In this thesis we studied three different applications of motion data and key-point trajectory analysis based on the different factors mentioned above.

The first application is analysing eye motion to better understand team cognition between members, specifically surgeons who formed a laparoscopic surgery team. Although team cognition is believed to be the foundation for team performance, there is no direct and objective way to measure it, especially in healthcare settings. In fact, the deficiency in tools for objective team assessment has been a major barrier in promoting surgical team training. Previous studies have shown that spatial features such as overlap analysis of eye-gaze data can be a measure of team cognition. However, due to the dynamic nature of eye-gaze signals, gaze overlap calculated from spatial features is not sufficient; as team members might look at the same surgical spot at different times. Therefore, temporal feature analysis is essential . Here we studied eye-gaze signals of two surgeons throughout a simulated laparoscopic surgery task and distinguished expert teams from novice teams by the level of gaze overlap, the lag and the Recurrence Rate (RR) between two surgeons based on dual eye-tracking evidences. The results obtained in this study support the hypothesis that the top performing teams are better synchronized, show higher eye-gaze overlap and RR, and therefore demonstrate better team cognition.

The second application of motion data analysis is human fall detection using a 2D video sequence. Automatic, real time fall detection techniques can improve the life quality for seniors and people with special needs, as falling down can be life threatening for these groups. Computer vision based fall detection systems require less infrastructure and is cheaper and more comfortable for users compared to smart floors or systems based on wearable devices. However, vision based systems can be less accurate and not fast enough if the set of features and detection algorithms are not selected properly or the size and generality of the training dataset does not cover different specifications. Acquiring a general training dataset that covers all the possible conditions is very challenging, especially for unknown surveillance regions, such as in smart houses. We proposed a robust and real time, vision based fall detection technique using only a single RGB camera. The proposed method can be applied at frame level and only uses two significant points, head and center of person. Experiment were performed on le2i fall detection dataset which is publicly available. The proposed technique can distinguish falling from everyday actions, such as sitting down and sleeping. The proposed method can also work

in different indoor environments with different lighting conditions.

The last application is extracting the animation skeleton directly from human model regardless of its initial position and orientation. This can be used to automatically animate any arbitrary 3D character, which has many applications in simulation and entertainment. Defining trajectory key-points for 3D characters without manual intervention remains a challenging problem that makes complete automation difficult. To animate an articulated 3D character, a rigging process is needed, during which an animation skeleton needs to be extracted from or be embedded into a 3D model. This tedious process is mainly done manually by expert animators. Most of the automatic rigging techniques proposed in the literature are not fully automatic nor pose invariant, i.e., a front facing model in neutral T-pose is required at the start in order to animate successfully. We propose incorporating robust skeleton based feature detection, combined with identification of various anatomical characteristics, to extract the desired key-points along with constraint parameters needed for automatic rigging.

Preface

The majority of this thesis has been published or are under review in peer reviewed journals and conferences. Chapter 2 presents the first work in spatio-temporal eye-gaze analysis to understand team cognition during a laparoscopic surgery. The work has been published in IEEE Engineering in Medicine and Biology Conference, International Conference on Smart Multimedia and ACM Symposium on Eye Tracking Research & Applications. Chapter 3 discusses the details of robust and real time vision based fall detection system. It shows good improvement based on time performance, accuracy, precision and recall and is under review in IEEE Transactions on Pattern Analysis and Machine Intelligence. Chapter 4 provides details and results of our anatomically based mesh segmentation method. The results have been published in IEEE Systems Man and Cybernetics Conference. Finally, Chapter 5 is the first effort in literature to automatically extract animation skeleton from a 3D model regardless of its position and orientation. The work has been published in IEEE International Symposium on Multimedia and European Association for Computer Graphics (Eurographics). I choose to use first-person plural throughout this thesis to honour the contributions of my advisers and collaborators on my various works.

Related Publication During PhD Study

- Nasim Hajari and Irene Cheng, *A Real-Time Fall Classification Model Based on Frame Series Motion Deformation*, IEEE Transactions on Pattern Analysis and Machine Intelligence (Submitted)

- Nasim Hajari, Wenjing He, Irene Cheng, Anup Basu and Bin Zheng, *Spatio-temporal eye gaze data analysis to better understand team cognition* , International Conference on Smart Multimedia 2018

- Nasim Hajari, Irene Cheng and Anup Basu, *Robust Human Animation Skeleton Extraction Using Compatibility and Correctness Constraints*, 2016 IEEE International Symposium on Multimedia (ISM), San Jose, CA, 2016, pp. 271-274.

- Nasim Hajari, Irene Cheng, and Anup Basu, 2016 *Transferring and animating a non T-pose model to a T-pose model.* In Proceedings of the 37th Annual Conference of the European Association for Computer Graphics: Posters (EG '16). Eurographics Association, Goslar Germany, Germany, 29-30.

- Nasim Hajari, Irene Cheng, Bin Zheng and Anup Basu, *Determining team cognition from delay analysis using cross recurrence plot*, 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Orlando, FL, 2016, pp. 3482-3485.

- Bin Zheng, Nasim Hajari, and M. Stella Atkins, 2016. *Revealing team cognition from dual eye-tracking in the surgical setting*, In Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications (ETRA '16). ACM, New York, NY, USA, 321-322.

- Nasim Hajari, Irene Cheng, Anup Basu and Guillaume Lavoué, *Evaluation of 3D Model Segmentation Techniques Based on Animal Anatomy*, 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, 2013, pp. 3277-3281.

To Hassan, Ryan and Kiana

For teaching me the most important concept in life.

Those who can imagine anything, can create the impossible.

– Alan Turing.

Acknowledgements

First, I would like to express my heart felt gratitude to my advisers Dr. Anup Basu and Dr. Irene Cheng for their continuous support and guidance throughout my PhD studies. Thank you for your patience, motivation and great supervision. It meant a lot to me and helped me develop critical thinking and research ability.

Second, I am grateful to Dr. Bin Zheng for his guidance and support for the team cognition project. Working closely with surgeons and the Surgical Simulation Research Lab (SSRL) opened a new door of research to me and helped me understand the problem better.

I would also like to thank the Department of Computing Science and NSERC for their financial support, which made this research possible.

I am also thankful to my friends at the University of Alberta: Housam, Nathaniel, Navaneeth, Megha, Sara, Amir and Mehdi for their support and friendship. Having you as friends made this journey joyful.

I would also like to thank my thesis committee for their encouragement, insightful comments, and thoughtful questions, which helped me become a better scientist.

I would also like to thank my parents for all their support, understanding and patience. Also, I would like to thank the joy of my life Ryan and Kiana. I found the meaning of life with you, and most importantly I would like to thank my husband as he is always there for me and been my best emotional support.

Lastly, I would like to thank each and everyone who has directly or indirectly been a part of my Ph.D. journey. Without your support this thesis would not be possible.

Contents

List of Tables

List of Figures

List of Terms

CNN Convolutional Neural Network

COM Centre of Mass

CRA Cross Recurrence Analysis

CRF Conditional Random Fields

CRP Cross Recurrence Plot

CRQ Cross Recurrence Quantification

FD Foreground Detection

FN False Negative

FP False Positive

GMM Gaussian Mixture Model

HMM Hidden Marcov Model

HOG Histogram of Oriented Gradient

LISETT Legacy Inanimate System for Endoscopic Team Training

MoCap Motion Capture

OSCVM One Class Support Vector Machine

PCA Principal Component Analysis

PD Person Detection

RA Recurrence Analysis

RC Random Cut

R-CNN Region Convolutional Neural Network

RNN Recurrent Convolutional Neural Network

ROI Region of Interest

RP Recurrence Plot

RR Recurrence Rate

RW Random Walk

SDF Shape Diameter Feature

SVM Support Vector Machine

TN True Negative

TP True Positive

VNSM Valance Normalized Spatial Median

WHO World Health Organization

Chapter 1

Introduction

Motion is defined as the action or process of moving or being moved. In physics any changes in the position of a human, animal or object over time is motion. Usually motion is described in terms of displacement, distance, velocity, acceleration, time, and speed.

There are different technologies to detect a moving object or human in the scene. This is a vital part of any security, surveillance or smart room applications. Passive infrared (PIR) sensors works based on human body heat and in the presence of a living being in the covered area they activated. The Microwave sensors emit microwave pulses frequently and measure the reflection of a moving object. The use of low cost digital 2D or 3D camera is also very common for these applications. Although, one should use customized software to detect motion from the recorded video or recognize different gestures. Some technologies combine different sensors to eliminate error and noise in the scene due to lighting, coverage area or other external factor. An example of such technology is Microsoft Kinect where an infrared sensor and a digital camera is combined to detect and recognize motion.

In computer science, motion analysis can refer to studying a few consecutive 2D images, from a captured video, to detect moving objects or recognize the actions in the video. In a 3D context it can also refer to capturing and studying the displacement of 3D humans, animals or object key-points. This process is called Motion Capture or MoCap. However, one can also use non-visual sensors to capture motion of different body parts, such as eyes, or other features related to motion, such as acceleration.

Analyzing and studying the trajectory or other motion related features of different key-points can reveal important information for different applications. For example, surveillance applications heavily depend on 2D video motion analysis and action recognition. Entertainment, games and healthcare applications can benefit from 3D gait and MoCap analysis. Even for psychological applications, such as studying individual cognition and team cognition, one can use eye motion data. It is even possible to combine all of these small applications together to create a smart simulation room, which can be used for education and training purposes. An example can be an smart simulation room in healthcare settings where different sensors such as eye-tracker, Kinect, leap motion sensors and regular or depth RGB cameras is used to capture key point trajectories of expert professionals. It is then possible to use this data to compare the deviation of trajectories between experts and novices and therefore correct their actions accordingly.

1.1 Thesis Scope

This thesis focused on three different motion analysis applications. The first application is analysing 1D eye motion to better understand team cognition between two surgeons. Section 1.1.1 talks about the motivation behind this study. The second application is fall detection in 2D videos. Section 1.1.2 explains the importance of fall detection. The last application is automatically detecting the 3D trajectory key-points of an arbitrary 3D model. Section 1.1.3 discusses more details for this application.

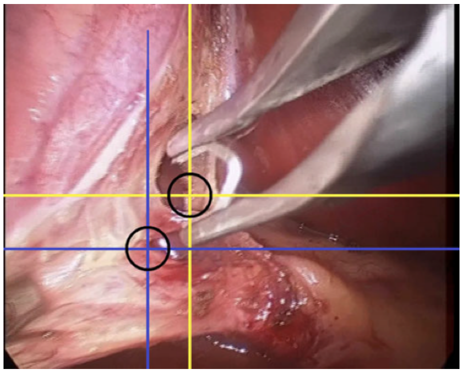


Figure 1.1: Dual overlay of eye-gaze data on a laparascopic surgery site.

1.1.1 1D Application: Eye Motion Analysis

The first application is 1D eye-gaze analysis of two team members during a laparoscopic surgery to understand team cognition.

The extensive benefits of minimally invasive surgery, such as shorter hospital stay and faster recovery time has led to an increasing demand in laparoscopic surgery for many abdominal procedures [10], [70], [102], [104]. For instance from 2013 to 2014, 28595 patients got their gall bladder removed, according to the data from the Canadian Institute of Health Information (CIHI) [91], and laparoscopic cholecystectomy has been rapidly becoming the routine choice for this procedure.

Laparoscopic surgeries are mostly performed by teams of two or more surgeons [19]. Salas and other authors defined a team as “a distinguishable set of two or more people who interact dynamically, interdependently, and adaptively toward a common and valued goal/object/mission, who have each been assigned specific roles or functions to perform, and who have a limited life span of membership” (p. 4) [82]. In a laparoscopic surgery team, for example, an assistant usually controls the primary surgeon’s vision by manipulating a laparoscope through the surgery site. As several studies suggest, having a relatively untrained assistant in the team may affect the primary surgeon’s course of action and decision making due to non-optimal display of the surgical site [34],

[39], [125]. These factors, further imposed upon the surgeon, who is already carrying considerable ergonomic difficulties related to laparoscopy, may eventually lead to overt surgical errors that compromise patient safety [16], [34]. Therefore, teamwork in laparoscopic surgery is crucial and needs to be explored. It has been shown that a trained laparoscopic team can achieve better results in terms of operation time, patient care and overall cost [56]. However, the available training programs mainly focus on individuals and surgeons are evaluated on an individual basis [30], [31], [80], [124]. Although team cognition is believed to be the foundation for team performance, there is no direct and objective way to measure it, especially in the healthcare setting. In fact, the lack of objective assessment tools has been a major barrier in promoting surgical team training [41], [94], [111].

In this thesis, we study the spatial feature of surgeons' gaze from elite and poor performance teams through overlap analysis. We also analyse the temporal features of dual eye tracking signals recorded simultaneously from two human operators using cross correlation and CRA algorithms. Considering both features enables us to reveal more reliable evidence for shared cognition of surgeons in laparoscopic surgery.

1.1.2 2D Application: Video Motion Analysis to Detect Fall

The second application is human fall detection by analyzing person movements based on two significant points, head and center of person, using a 2D video sequence.

The life expectancy of seniors has been increased due to advancements in health care services. As World Health Organization (WHO) [119] and Carone and Costello [18] suggest, the world population over 60 years old will double by 2050, which will be a total number of 2 billions. It is obvious that all countries will face the impacts of this huge growth and should change their health and social systems accordingly. The main need for senior population is living safely and independently either in their homes or in assisted care facilities. The major incident with severe physical and emotional impact on an elderly is accidental fall [45]. It can even be life threatening if the person cannot seek help on time. Therefore designing and implementing fast and reliable fall detection methods are getting more attention during the past decade.

Computer vision based fall detection systems require less infrastructure and is cheaper and more comfortable for the users compared to the systems using smart floors or wearable devices. Usually a set of features are extracted and processed for each video sequence to detect fall. However, the vision based systems can be less accurate and not as fast as wearable devices if the set of features and detection algorithms are not selected properly. Also, for any learning based applications the size and generality of the training data plays an important role. If the training dataset is not comprehensive enough, and does not cover different viewing direction or illumination condition, the fall detection model will not work for real scenarios.

Acquiring a general training dataset that covers all the different conditions is very challenging, especially if the surveillance area is not known beforehand. In this thesis we propose a robust and real-time, vision based fall detection technique using a single RGB camera only.

1.1.3 3D Application: Automatic Detection of 3D Trajectory Key-points

With the advances in animation technology and demands of graphics applications, 3D models as well as MoCap data are being created at a rapid rate both in number and variety. By associating a 3D model with different MoCap data, a variety of appealing animations can be generated. Most applications

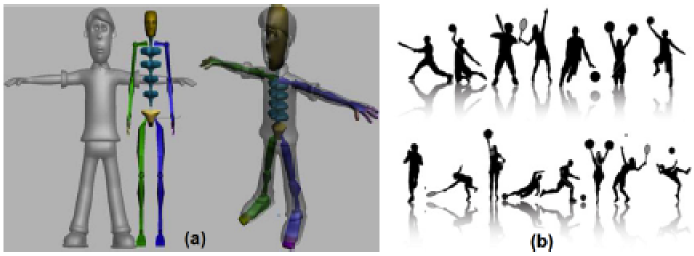


Figure 1.2: (a) Manual rigging with the help of commercial software is tedious, and it is not easy to link the skeleton with the desired MoCap data and deliver good animation. (b) Restricting the rigging on a T- or A-pose model is not practical because majority of the animations start from other poses.

in entertainment, training and simulation require 3D human (biped) models in the form of a 3D mesh. The main challenge for application developers is to efficiently and accurately link the desired MoCap data to a 3D skeleton, generated from the mesh, in order to produce a realistic animation. To satisfy consumer demand, built-in tools are available in commercial software, e.g., AutoDesk 3DsMax and Maya, to support such a process (Fig. 1.2(a)) so that novices without graphics programming skills can participate in animation production. Despite the support of these expensive commercial softwares, using these tools still involves tedious manual effort (i.e., setting constraints, kinematics, bone hierarchy and so on) with no guarantee of delivering satisfactory animation result. To address this issue, many state-of-the-art rigging techniques have been introduced in recent years.

Before linking the MoCap data to a 3D skeleton, the skeleton itself has to be created [13]. Two approaches are commonly used for skeleton creation: (1) extracting a unique skeleton for each 3D model, or (2) starting from a generic skeleton, adjusting and rigging it to fit into individual 3D models. The advantage of the latter approach is that a single template skeleton is sufficient for multiple models. However, the limitation is that for arbitrary models that are not upright, front facing and in a neutral position, the animation can be funny and unrealistic as shown later in this paper. While the former approach requires a skeleton generation step, it deals with a model specific skeleton and thus has the advantage of choosing more appropriate key-points to match different MoCap data. Also, the generated skeleton provides information on model orientation.

An important step for skeleton-based animation is rigging, which means embedding a skeleton inside the 3D character and associating the mesh vertices to the skeleton bones. Motion retargeting is the next step, which means adapting the animated motion from one articulated character to another. The final step is skinning, where the mesh vertices are attached to the skeleton bones controlled by the assigned weights.

Despite recent advances in rigging algorithms, they are mostly not fully automatic and have limitations, e.g., a technique may require a front-facing, A- or T-pose 3D humanoid model as the starting position. Adjusting individual characters is labour intensive and is not feasible for real-time applications.

To address these limitations, we present two new techniques to automatically extract an animation skeleton for an arbitrary 3D model, regardless of its position and orientation.

Chapter 2

1D Eye Movement Analysis to Understand Team Cognition

Motion analysis, as mentioned before, has a broad range of applications. There are different technologies to capture the motion of different body parts such as eyes. Eye-gaze devices are used for such purposes. Eye motion is an strong indication of cognitive function as described by [40].

Studying and understanding team performance is very important for sports, games, health and applications that involve a team of users. Team performance is affected by team behaviour or cognition. Usually a team with a good shared cognition can perform better and achieve the set goal faster.

Having a good team with a good shared behaviour is even more crucial in health care environments, especially in surgery. Surgery is a team effort and shared cognition between surgeons in a team is pertinent for operation quality and patient safety. Analyzing team cognition during minimally-invasive surgery is a new area of research.

In this chapter, we analyze the 1D eye motion data to understand the team cognition between two surgeons, who perform a simulated laparoscopic task. The eye motions of two surgeons were recorded during a simulated surgical operation. We then performed spatio-tempral analysis such as Cross Recurrence Plot (CRP), Cross Recurrence Analysis (CRA), lag analysis and overlap analysis to find spatio-temporal features that can be used to distinguish between a good and a bad performing team.

Dual eye tracking data for twenty two dyad teams were recorded during the simulation and then the teams were divided into good performer and poor performer teams based on the time to finish the task. We then analyse the signals to find common features for good performer teams. The results of this research indicates that the good performer teams show a smaller delay as well as have a higher overlap in the eye-gaze signals compared to poor performer teams.

2.1 Background and Related Work

Teamwork is critical in many work environments such as sports and healthcare. However, current assessment tools for teamwork are based on subjective assessments. Minimally invasive surgery is one application of teamwork, which is getting more popular. It has extensive benefits over conventional surgery such as shorter hospital stay and faster recovery time. Laparoscopic surgeries are mostly performed by teams of two or more surgeons [19]. In a laparoscopic surgery team, an assistant usually controls the primary surgeon's vision by manipulating a laparoscope through the surgery site. As several studies suggest, having a relatively untrained assistant in the team may affect the primary surgeon's course of action and decision making due to non-optimal display of the surgical site [34], [39], [125]. These factors, further imposed upon the surgeon, who is already carrying considerable ergonomic difficulties related to laparoscopy, may eventually lead to overt surgical errors that compromise patient safety [16], [34]. Therefore, teamwork in laparoscopic surgery is crucial and needs to be explored. It has been shown that a trained laparoscopic team can achieve better results in terms of operation time, patient care and overall cost [56]. However, the available training programs are mainly focused on individuals, and surgeons are evaluated on an individual basis [30], [31], [124]. Although team cognition is believed to be the foundation for team performance, there is no direct and objective way to measure it, especially in the healthcare setting. In fact, the lack of objective assessment tools has been a major barrier in promoting surgical team training [41], [94], [111].

One method to assess team collaboration is video analysis. [127] analysed a simulation of an endoscopic cutting task performed by two operators. This study showed that team performance is highly correlated with team collaboration. In recent years people used eye tracking systems to study different features of health care providers. A procedural and objective method to assess a surgeon's skill is through eye gaze analysis and reporting the level of gaze overlap. However, team members may look at the same location at different time points. To understand team cognition better, one needs to consider both spatial and temporal features of eye gaze data. Therefore, in this chapter, we study the spatial features of surgeons' gaze from elite and poor performer teams through overlap analysis. We also adopt CRP algorithms and lag analysis to analyse the temporal features of dual eye tracking signals recorded simultaneously for two human operators. Using both features, one can reveal more reliable evidence for shared cognition of surgeons in laparoscopic surgery. We hypothesize that there is a relationship between task completion time, recurrence rate, time delay and overlap between eye gaze signals for good performance team and bad performance team during the simulated surgery task. This is helpful in developing a novel method for assessment of the level of team cognition.

In this section we discuss the related work in the area of eye tracking and team cognition as well as CRP and team cognition.

2.1.1 Eye Tracking and Team Cognition

Eye tracking technique, as an objective assessment of surgical skills has been well documented in the literature [6]. Gaze patterns have been shown to differentiate poor and elite surgeons in several studies [33], [57], [108]. Also, eye tracking can be used to examine the workload and vigilance of surgeons [109], [128]. Video analysis of an endoscopic cutting task performed by one vs. two operators indicates that good team collaboration results in superior team performance [127], and higher frequency of anticipatory movement was noticed in dyad teams [126]. Later on, Khan and Zheng [57] used dual eye-tracking to examine the spatial similarity in eye-tracking between two surgeons. Reporting level of gaze overlap is an innovative step in the study of shared cognition between two surgeons in a laparoscopic team. One problem is phase difference and delay between team members, which can be solved by lag analysis or CRP and CRA analysis.

2.1.2 CRP and Team Cognition

Cross Recurrence Quantification (CRQ) is a useful statistical tool for dynamic systems and useful to find relation or interrelation between two time series. It can quantify the similarity between two time series unfolded over time. One value of such quantification is recurrence percentage, which describes how often two series go through similar system states. CRQ is a reliable tool for dynamic, short and complex data series. It is a very useful tool to study human interaction over time. Linear data analysis is not suitable to analyse the short, non-stationary and complex data series. An appropriate method to analyse this type of data is Recurrence Plots (RP). It has been proven that recurrence is a fundamental property of dynamic systems, which means that after some time the system will reach the state that is arbitrarily close to the former states and pass through a similar evolution. RP can visualize the recurrence behaviour of dynamic systems. Also, one can perform the Recurrence Analysis (RA) based on the RP and calculate the Recurrence Rate (RR). Eq. 2.1 shows RR as it is explained in [73].

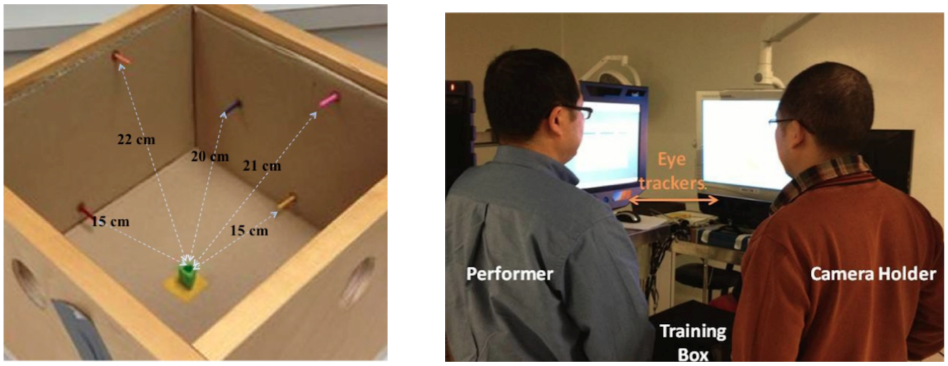
$$RR = \frac{1}{N^2} \sum_{i,j=1}^{N} \boldsymbol{R}_{i,j} \tag{2.1}$$

where N is the number of points on the phase space trajectory, i and j belong to the two different data series that we are studying and eventually $\boldsymbol{R}_{i,j}$ is the RP as defined by Eq. 2.2.

$$\boldsymbol{R}_{i,j} = \Theta(\epsilon_i - \|\overrightarrow{x_i} - \overrightarrow{x_j}\|) \tag{2.2}$$

where $\overrightarrow{x_i}$ and $\overrightarrow{x_j}$ are the phase space trajectories of time series i and time series j respectively. Θ is the Heaviside function and ϵ is the threshold. The states of a natural or engineering dynamic system usually change over time. The state of a system x can be described by its d state variables, $x_1(t), x_2(t), ..., x_d(t)$. The vector $\vec{(}x(t))$ in a d-dimensional space is called phase space. The system's evolving state over time traces a path, which is called the phase space trajectory of the system.

In 2005, Richardson and Dale first used CRP to analyze gaze similarity recorded from two different persons [90]. They studied the relationship between a speaker and a listener based on their eye movements, and found that the coupling between a speaker's and a listener's eye movements indicate if the listener is engaged to the speaker or not. While the gaze movement of the speaker was recorded, he watched a television show and at the same time talked about it. Later, the listener watched the same show as he was listening to the previously recorded monologues and his gaze movements was recorded too. Finally, CRA was used to detect the matching behavior between speaker and listener‘s gaze movements. Marwan et al. presented a comprehensive re-



(a)

(b)

Figure 2.1: (a) The zoom in of the box setup(b) The experimental setup.

view on different CRP and CRA approaches [73]. One can find an excellent MATLAB toolbox or an R package [25] to perform CRP analysis.

In this application we used both RR and cross correlation to study the delay between two eye-gaze signals.

2.2 Experimental Setup

This section explains the experimental setup and data collection procedure. The study was performed in the Surgical Simulation Research Lab at the University of Alberta. Methods used in this experiment were reviewed and approved by the Health Research Ethics Board of the University of Alberta. Consent was obtained from each participant before entering the study.

2.2.1 Participants

Participants included 17 university students, office staff, and visiting scholars, of which none have received special training on laparoscopic surgery. They were asked to form 22 paired teams to perform a simple object transportation task under the simulated surgical environment using the laparoscopic technique.

2.2.2 Apparatus

The experimental set up includes four main components. The first one is a laparoscopic training box measuring 30 x 30 x 20cm (Fig. 2.1(a)). Inside the box, the distance of home plate to different pins is labeled (Fig. 2.1a), the setup of the simulation model is based on the Legacy Inanimate System for Endoscopic Team Training (LISETT) [80]. The training box has ports of entry for a 0-degree laparoscope and two laparoscopic grasper. The second component is two 17" video monitors (Tobii 1750 LCD Monitor, Tobii Technology, Stockholm, Sweden), which displays the image captured by a laparoscope and a webcam. We also used a standard laparoscopic imaging system, including laparoscope, camera, light source and video monitor (Stryker Endoscopy, San Jose, California, USA). Finally, two high-resolution remote eye-trackers (Tobii 1750 and X50, Tobii Technology, Stockholm, Sweden) were set in an orthogonal arrangement (Fig. 2.1(b)). Each eye-tracker can remotely track one operators' eye motions unobtrusively within a comfortable viewing distance. Gazes of two operators in a dyad team were recorded separately by the two eye-trackers and the data fed into the Labview software to synchronize the gazes in time on top of the surgical video streams.

2.2.3 1D Signal formation

Before further processing of the captured data, we should clean the data and make it usable. This preprocessing step involves filtering the data to remove the noise and interpolate to fill out any discontinuity. It is obvious that the data we gathered from eye trackers is 2D, which are the x and y coordinates of the eyes fixation point. However, for the sake of simplicity and better understanding of the problem, and in order to use the CRP package we projected the 2D data into a 1D space. Here we tried two different distance metrics, Euclidean and Manhattan distance to perform the projection. Note that the

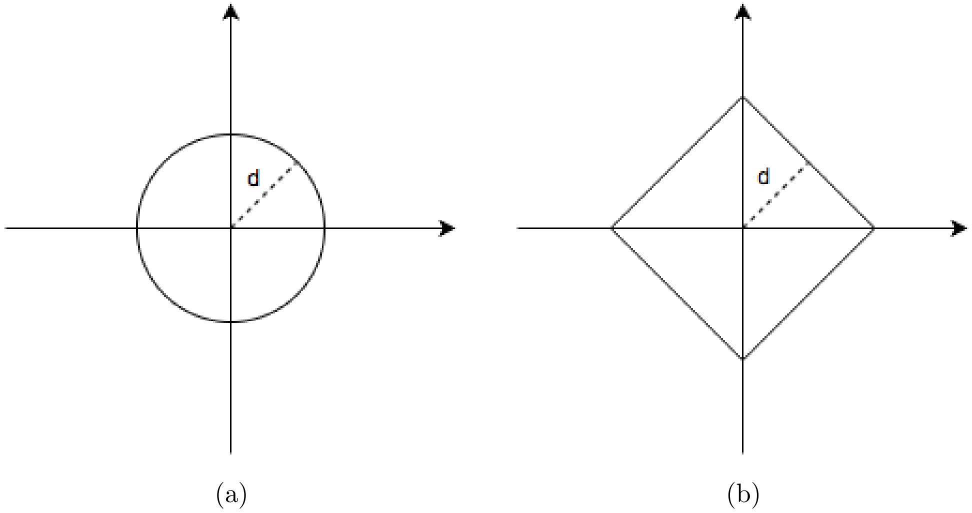


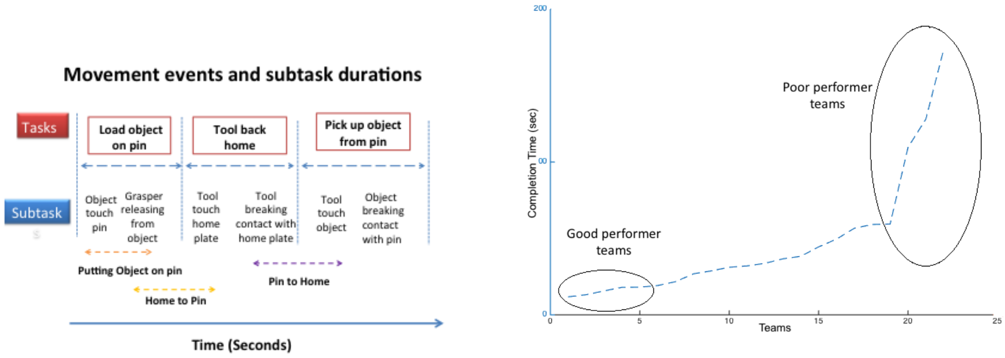
Figure 2.2: (a) Distribution of Euclidean distance on a circle(d) Distribution of Manhattan distance on a diamond.

x and y coordinates of the eye's fixation point are not necessarily correlated and therefore applying Principal Component Analysis (PCA) to reduce dimensionality is not a very good idea. Also, any sort of projection to reduce dimensionality will introduce some error into the system.

The distance metrics calculate the distance between eye's fixation point and the origin and is represented by d. For the Euclidean metric d equals $\sqrt{x^2 + y^2}$, while for the Manhattan metric or L-1 norm d equals $|x| + |y|$. This implies that for Euclidean distance all the points that lie on a circle will have the same distance (the radius of the circle) from the origin; while for Manhattan distance all the points that lie on a diamond with four equal sides have the same distance from the origin. Fig. 2.2 shows these conditions for both metrics.

2.2.4 Procedure and Tasks

Each dyad team was asked to perform a laparoscopic procedure, which consisted of a set of tasks. The camera holder was required to navigate a 0-degree laparoscope to locate different coloured pins for the primary performer. The



(a) (b)

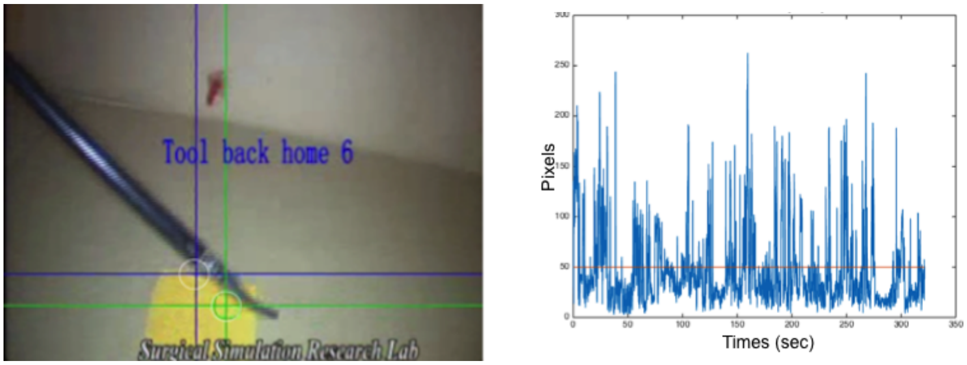
Figure 2.3: (a)Procedure and task description; (d) Completion time for the whole task of the orange pin for 22 teams.

performer then executed the task of grasping and transporting a plastic cylinder (2 cm long, 1.5 cm wide) among the pins. Fig. 2.3(a) explains the steps required during each task. The pins, which were 2 cm in length and protruding out of the center, were mounted on two interior side-walls of the wooden box (Fig. 2.1(a)). In total, there were five different coloured pins (blue, red, orange, pink, and yellow). The experimenter randomly assigned the sequence of selecting the coloured pins. The camera holder needed to manipulate the laparoscope forward and backward to locate pins, the object and the home position. (S)he was also required to rotate the light cord clockwise and counterclockwise to keep the object and the instrument at the center of the view, and adjust the focus of the camera to provide a clearer image.

2.3 Measurement and Data Analysis

We analyzed the elite and poor teams based on the spatial feature, which is gaze overlap percentage, as well as the delay between two team members using cross-correlation and CRP analysis. We also reported the recurrence rate and cross-correlation between two team members gaze signals.

To get a distinctive category of elite and poor performer teams, we chose the



(a)

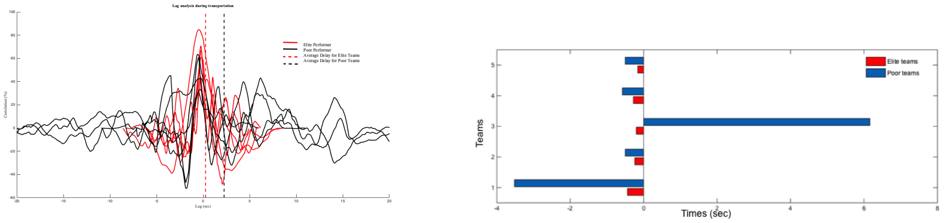
(b)

Figure 2.4: (a) The overlay on dual eye tracking data on one frame of the video; (b) The eye-gaze overlap analysis.

top and bottom 25% of the teams based on the completion time of the orange pin's task, which is shown as two selected regions in Fig. 2.3(b). For overlap and delay analysis, we mainly focused on the tool transportation period. We specifically considered the period in which the tool transports from the orange pin to home and from home to the orange pin. The orange pin is chosen because of its greater distance to home plate; therefore, providing sufficient space to observe collaborative behaviours. Note that both the orange and pink pins have the greatest distance from the home plate. However, we chose the orange pin because the separation between good performer and poor performer teams is more visible for this pin. The following subsections describe the spatial, temporal and delay analysis in more detail.

2.3.1 Spatial Data Analysis (Gaze Overlap)

We calculated the pixel overlap of the camera driver and the performer to study the spatial features. [57] have shown the level of gaze overlap is highly correlated with the level of expertise. Previous studies [57], [108] suggested that a visual angle of at least 3° is required to mark eye gaze data as mismatch. Therefore, we set the threshold to 50 pixels, which indicates a gaze separation of almost 5° visual angle for our setup and resolution. In other words if



(a)

(b)

Figure 2.5: (a) Distribution of phase delay between good performers and poor performers; (b) The delay distribution for good performers and poor performers.

the Euclidean distance between the location of two gaze signals is smaller than 50 pixels then both members are almost looking at the same location. The white circle in Fig. 2.4(a) demonstrates that the overlapping area in the simulation environment and the red horizontal line in Fig. 2.4(b) shows a 50 pixel separation threshold on the difference between two gaze signals (blue curve).

2.3.2 Cross-correlation and Delay Analysis

We used cross correlation to calculate the delay between team members. Cross-correlation of two signals $X = (X_t)$ and $Y = (Y_t)$ is the function that gives the correlation of the two signals at different time points. It shifts one signal and keep the other signal fixed. Note that cross-correlation of two signals is similar to convolution of two functions. It is used as a measure of similarity between two signals. Also, it can detect if two signals have a lag relative to each other for the time delay analysis. Eq. 2.3 shows how to calculate cross-correlation.

$$(X \star Y)(\tau) = \int_{-\infty}^{\infty} X^*(t)\; Y(t+\tau)\, dt, \tag{2.3}$$

Where X^* is the conjugate of signal X. The maximum cross-correlation between the two signals is the point in time where two signals are best aligned. This represents the delay between the two signals. Fig. 2.5(a) shows the distribution of the phase delay for good performer and poor performer teams as well as the energy level. As the dotted lines show the average delay for good performer teams are much lower compared to the poor performer teams.

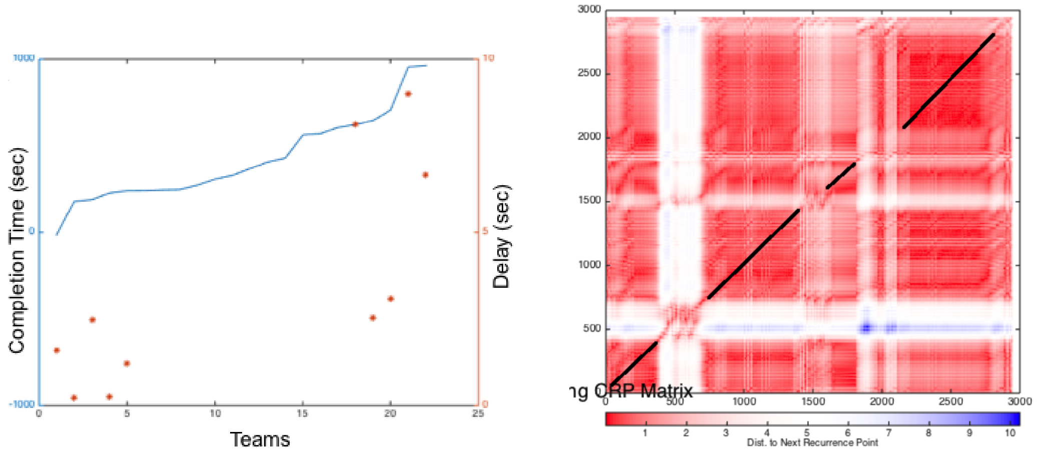
Another factor we can compare between two groups is the maximum energy for cross-correlation plots. On the average, the energy of good performer teams is higher compared to the poor performer teams. The energy is the magnitude of the peak of the cross-correlation curve. Note that it is a normalised cross-correlation with values between $[-1, 1]$. The vertical axis of Fig. 2.5(a) shows the percentage of this energy value. The average energy for good performer teams is 74% while the average energy for poor performer teams is 65%.

Fig. 2.5(b) shows the delay distribution. Note that a negative delay means that the camera holder is ahead of time, which is a characteristic of a good performer team and is an indication that they are expert surgeons. However, in the poor performer teams the positive average delay means the performer gaze signal is ahead of time. In other words, the camera holder is a novice.

2.3.3 Recurrence Rate and Delay Analysis

Another way to calculate the lag between two team members is through RR as explained in Eq.2.1 and used by [90]. We calculated RR for different time delays between the two signals and the gap that generated the highest RR is selected as the delay between the two team members. The orange dots in Fig. 2.6(a) shows the corresponding delay for good performer and poor performer teams based on RR. As it is expected, good performing teams show smaller delay compared to poor performing teams. Note that here the total completion time for all the pins is presented. Also, the selection of good and poor performing teams based on the orange pin completion and total completion times are closely related.

Fig. 2.6(b) shows the main diagonal line for the recurrence patches in CRP



(a)

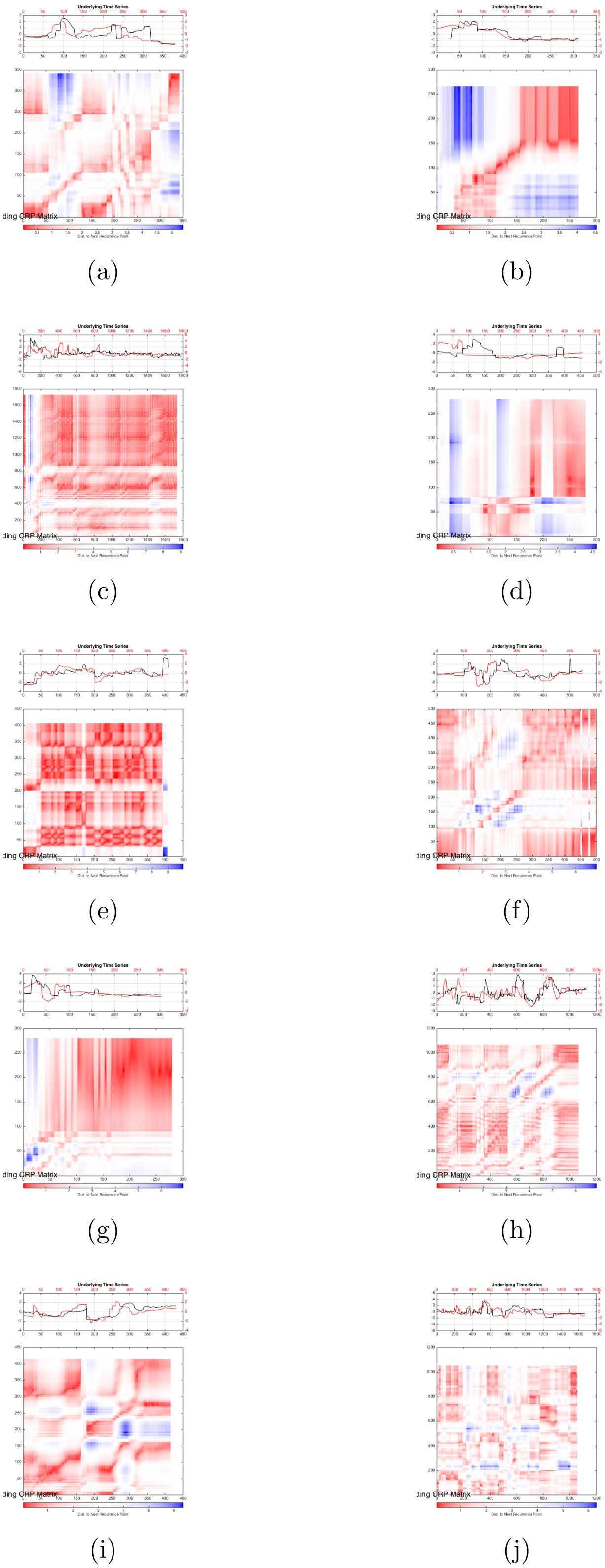
(b)

Figure 2.6: (a) The overall task completion time in blue and delay distribution in orange for top 25% and low 25% teams (based on the completion time of the orange pin); (b) An example of RR in CRP shown in black.

plots, which represents the recurrence rate.

2.3.4 CRP, CRA and Team Cognition

The first column of Fig. 2.7 and Fig 2.8 shows CRP for elite teams and the second column belongs to poor performer teams for the overall procedure and tool transportation period for the orange pin, respectively. The diagonal area corresponds to the recurrence rate. As these figures demonstrate, the plots for elite teams reveal more recurrence area and also appear to be denser and more clustered, while the plots for the poor performer teams are more random and do not show the overlap patches very well. We also used CRA to get the numerical value for recurrence rate for both elite and poor performer teams. The results are presented in Fig. 2.9(a). The overall recurrence rate for elite teams is higher compared to the poor performer teams.



(a) (b)

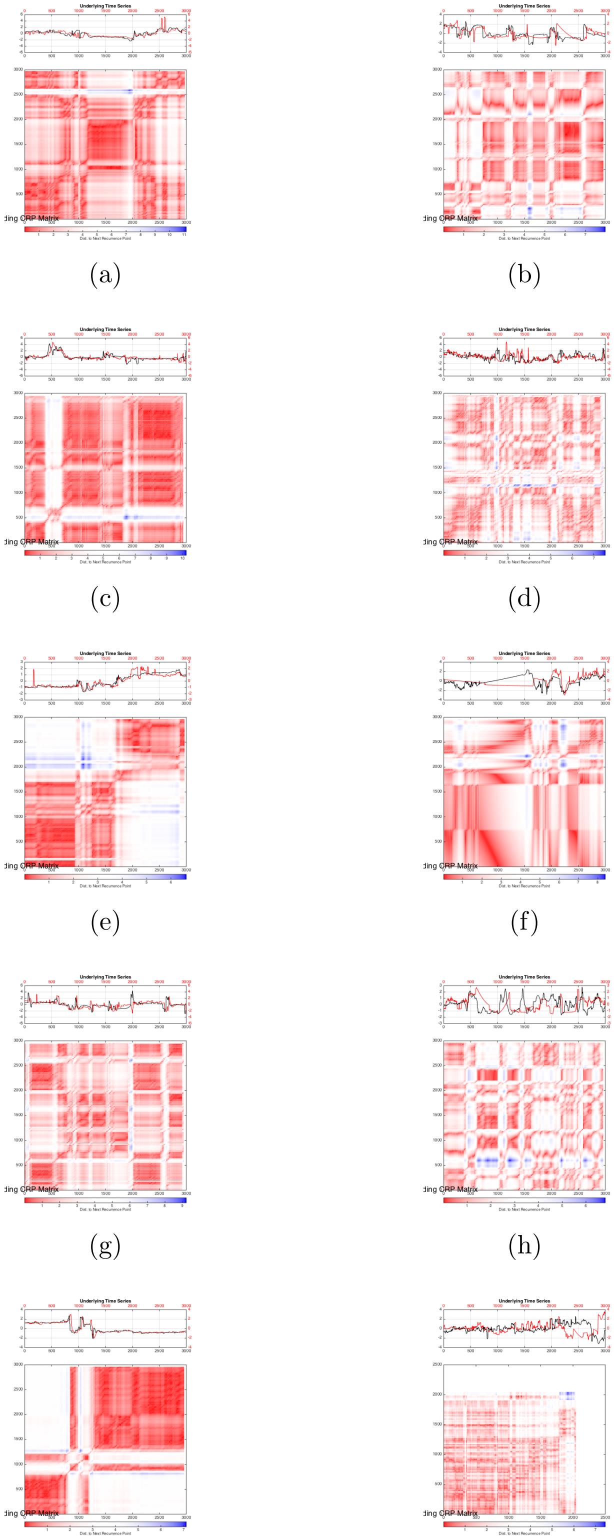
(c) (d)

(e) (f)

(g) (h)

(i) (j)

Figure 2.7: The cross recurrence plots for elite (first column) and poor performer (second column) teams during tool transportation period for the orange pin.



(a) (b)

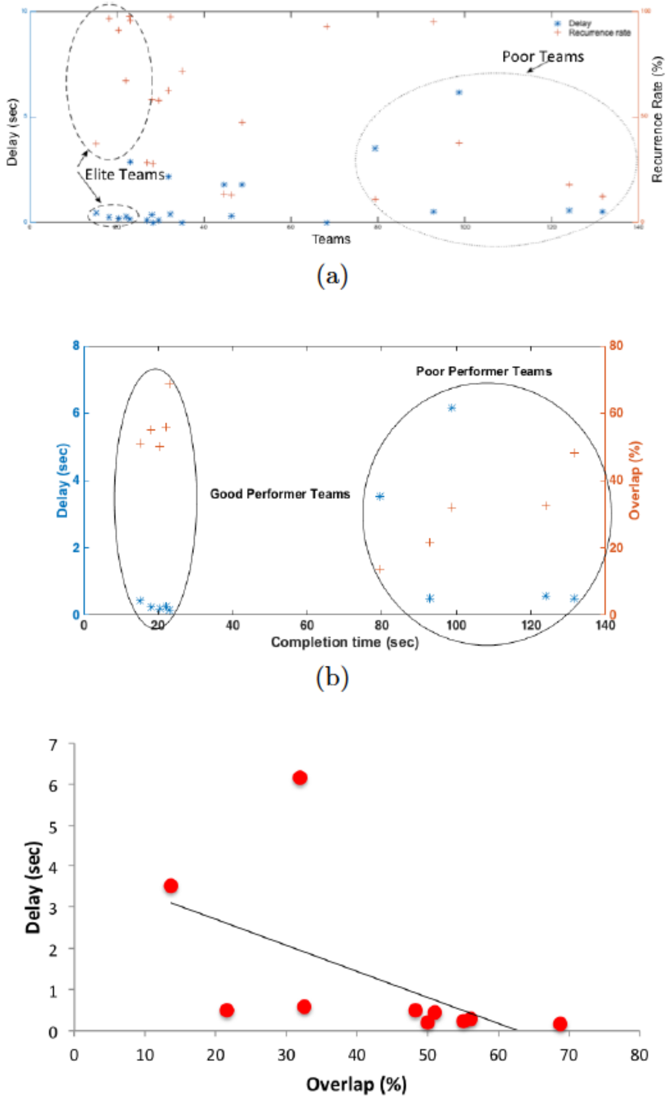
(c) (d)

(e) (f)

(g) (h)

(i) (j)

Figure 2.8: (a,c,e,g,i) Overall Cross Recurrence Plot for good performer teams; (b,d,f,h,j) Overall Cross Recurrence Plot for poor performer teams.



(a)

(b)

(c)

Figure 2.9: (a) The recurrence rate and delay distribution for poor performer and elite teams. (b) The gaze overlap (orange cross) and delay distribution (blue star) for good performer and poor performer teams.(c) The delay versus overlap and its trend line.

2.4 Results

In this subsection we present the results of both spatial and temporal data analysis.

2.4.1 Spatial Analysis

Table 2.1 shows the gaze overlap percentage for elite and poor performer teams during the whole procedure and orange-pin tool transportation periods. The average total gaze overlap between two team members in the elite team is higher than the poor performer team (Elite: 35.87 ± 4.84%; Poor: 28.74 ± 6.34%; P = 0.018), while the average transportation overlap for elite teams is $50.97 \pm 9.22\%$, significantly higher than the poor performer teams ($29.56 \pm 18.15\%; P = 0.023$).

2.4.2 Temporal Analysis

The average delay for elite and poor performer teams is presented in Table 2.1. CRA reveals a higher recurrence rate between two team members for elite teams (78.06 ± 25.93%) than poor teams ($34.41 \pm 34.42\%; P = 0.0412$). Further analysis show that two team members in poor teams displayed a 2.25 ± 2.54 sec gaze delay; whereas the delay dropped to 0.26 ± 0.11 sec for elite teams; $P = 0.032$. Also the camera holder leads the performer in the elite teams while in the poor performer teams the performer leads the camera holder on average.

2.4.3 Conclusion and Future Work

Understanding team cognition in a healthcare environment by analyzing a surgeon's eye-gaze data is a new area of research. Although team cognition is believed to be the foundation for team performance, there is no direct and objective way to measure it, especially in the healthcare setting. In fact, the

Table 2.1: Comparison of different features for elite and poor performer teams

	Total Overlap (%)	Trans. Overlap (%)	Overall Delay (sec)	Tool Trans. Delay (sec)	Recurrence Rate (%)
Elite Teams	35.87 ± 4.84	50.97 ± 9.22	1.78 ± 1.06	0.3 ± 0.12	78.06 ± 25.93
Poor Teams	28.74 ± 6.34	29.56 ± 18.15	6.06 ± 3	2.27± 2.94	4.41 ± 34.42
P value	0.018	0.023	0.032	0.029	0.0412

deficiency in tools for objective team assessment has been a major barrier in promoting surgical team training.

Previous studies showed that spatial features of eye gaze signals, such as overlap analysis, can be a measure of team cognition. However, due to the dynamic nature of the eye-gaze signals, gaze overlap calculated from spatial features is not sufficient, and temporal features of gaze signals should be analyzed too. The intuition is that team members might scan over the same surgical spot at different time steps. The CRP and CRA allows us to capture this temporal feature. Therefore, we believe they provide a more powerful tool for spatio-temporal analysis and refer better to shared cognition than the gaze overlap. The results presented in this thesis support our hypothesis that the top performance team, which is an indication of better team cognition, displayed higher recurrence rate (Fig. 2.7, Fig. 2.8, Fig. 2.9(a)). Specifically, two members in good performer teams scanned over the same surgical spot almost simultaneously, whereas members in the poor performance teams failed to scan over the same surgical spot at the same time. One of them, often the camera holder is behind the operator. Generally, the delay would be higher and recurrence rate and overlap would be lower for teams with longer completion time.

This study analysed the teams based on completion time. However, the design of the study should include expert surgeons in the elite teams, and compare their performance to teams comprised of novice surgeons. We plan to perform a new study by including surgeons with different level of surgical expertise. Based on the results of our study, dual eye-tracking and CRP/CRA is demonstrated to be a powerful tool for revealing temporal dependencies between team members which can be an indication of team cognition.

Chapter 3

2D Human Motion Analysis to Detect Fall

Motion analysis is an important part of any surveillance or action recognition application such as fall detection. Falling down is a dangerous and life threatening for seniors or patients with severe medical conditions such as Parkinson, especially if the person is alone and cannot seek immediate assistance. Therefore, automatic, real time fall detection techniques can improve the life quality for these groups and help them maintain their independence.

Computer vision based fall detection systems require less infrastructure and is cheaper and more comfortable for users compared to systems using smart floors or wearable devices. Usually a set of features are extracted and processed for each video sequence to detect fall. However, the vision based systems can be less accurate and not as fast as wearable devices if the set of features and detection algorithms are not selected properly. Also, for any learning based applications the size and generality of the training data plays an important role. If the training dataset is not comprehensive enough, and does not cover different viewing direction or illumination condition, the fall detection model will not work for real scenarios.

Acquiring a general training dataset that covers all the different conditions is very challenging, especially if the surveillance area is not known before hand.

In this chapter we propose a robust and real time, vision based fall detection technique using only one RGB camera. The proposed method works at the frame level and only use two significant points for classification, which makes it quite robust and real time. Experiments are performed on le2i fall detection dataset which is publicly available. It shows that the proposed technique can distinguish falling from everyday action such as sitting down and sleeping. The proposed method can also work in different indoor environments with different lighting conditions.

3.1 Background and Related Work

The life expectancy of seniors has been increased due to advancements in health care services. As World Health Organization (WHO) [119] and Carone and Costello [18] suggests, the world population over 60 years old will double by 2050, which will be a total number of 2 billions. It is obvious that all countries will face the impacts of this huge growth and need to change their health and social systems accordingly. The main need for seniors is living safely and independently either in their homes or in assisted care facilities. The major incident with severe physical and emotional impact on an elderly is accidental fall [45]. It can even be life threatening if the person cannot seek help on time. Therefore, designing and implementing fast and reliable fall detection methods are getting more attention during the past decade.

Fall detection approaches can be classified into three main groups, based on the underlying infrastructure [81]. The first group of techniques use wearable devices [21], [86]. Sensors such as accelerometers and gyroscopes are either embedded in the clothes or some specific devices such as smart watches. As the sensors are in close contact with the person they can produce very accurate results. However, they can be ineffective if the user forgets to put them on or if for some reason, such as a dead battery, the device stops working. Thus, it is not always a proper solution for elderly people.

The second group of fall detection techniques use ambient devices, such as pressure sensors on the floor [58], [129]. It is quite costly to modify and add the sensors to the surveilance area. Also, it can easilly generate false positives [81].

The last group of techniques use computer vision based approaches. These techniques can use a single RGB camera, multiple cameras or depth cameras such as Kinect and extract a set of different features to detect fall [81]. More details on vision based techniques are presented in Section 3.2. The vision based approaches have several advantages over the previous two technologies. The hardware required is quite inexpensive and easy to set up. It is mainly a camera or a set of cameras for the whole monitoring region. Also users do not need to wear any extra clothes or devices. However, most people do not like the idea of closed circuits camera and human surveillance in their living environment because of privacy issues. A solution to this problem is automatic online fall detection techniques, without visual identification e.g. via, image or video, disclosed to unauthorized personnel. In such systems the alarm will activate if fall is detected automatically and the person in danger can receive the required help on time.

This chapter presents a vision-based fall detection technique using a single RGB camera. The proposed method extract significant motion features for each frame and classifies the frames into fall and no-fall action. The technique is suitable to monitor rooms in a senior residence. The outline of this Chapter is as follow: Related work on vision-based fall detection techniques are presented in Section 3.2. Details of the proposed method are explained in Section 3.3. Section 3.5 discusses the experiment setup and shows results for the proposed method. The last section concludes the chapter.

3.2 Literature Review

Person detection and tracking, pose estimation and action recognition is an established research area in the field of computer vision. Some of the famous methods are person detection based on HOG features (Histogram of Orientation Gradient) [27] and Pictorial Structure [4]. Almost all of the previously proposed person detection techniques detects people in an upright position. This is why detecting fall is quite challenging using these approaches and people proposed different techniques specifically for fall detection.

Vision-based fall detection techniques can be classified into two different groups. Some techniques used a single RGB camera to capture the surveillance area. While the second group of techniques used a depth camera or multiple cameras to capture the surveillance area. Almost any vision based techniques analyze the bounding box of the moving person, extract some features accordingly and then classify the action based on the extracted features.

Rougier et. al. [93] analysed human shape deformation through a video sequence by studying the person's silhouette throughout the motion. They used a shape matching approach to track the silhouette of the person. The idea behind their approach is that during a fall the changes in human silhouette will occur rapidly. However, those fast changes can happen for other fast actions, such as exaggerated jumping. They later classified the action into normal and abnormal activities using a Gaussian Mixture Model (GMM). They mentioned in their paper that this approach could run at 5 frame/sec, which is not adequate for a real time surveillance application.

The fall detection method proposed by [123] is also based on a single camera. They analysed the silhouette of the person by subtracting the background using a code book model. This implies the need of a sufficiently large dataset for the training phase in order to detect foreground accurately. For surveillance applications, specially for smart home applications, this is not feasible due to different combinations of different objects. They described the posture of the person in a 2D environment using fitted ellipse and shape structure. They used One Class SVM (OCSVM) to identify outliers, which is fall, from normal actions. Based on their extracted features, it is very challenging to distinguish falling down from sleeping. However, OCSVM is an unsupervised technique and is a good choice for outlier detection.

Another single camera approach is proposed by [77]. They incorporated the velocity feature as well as the posture features, so that the detection can be more robust. They used shadow changes as a sign of velocity. However, the extracted velocity can be very noisy. For instance if shadow of the person is not visible due to the lighting setup, or if the small changes in environment (such as illumination changes) is detected as shadows.

Method proposed by [97] needed a clear model with minimal occlusion of the background to perform background subtraction. They used features that describe the position and velocity. Head detection is an important part of their feature selection. They estimated the head based on the position and orientation of the moving bounding box. This can be very noisy if the person is carrying an object, such as a chair, where the width and height of the bounding box can change.

Unlike previous methods that directly studied the silhouette and bounding box of the moving object, [24] identified three important points for each person and then based on the position and orientation of the points, they detected fall. However, to get these points they need to detect the foreground accurately with no noise. The other problem with this approach is the lack of velocity.

People mainly used Kinect as a depth camera [3], [17], [35], [61], [76], [122] to get the 3d position of human body parts or joints. Several challenges are involved with Kinect cameras, such as extreme sensitivity to lighting condition, capture range and the noisy skeleton generated.

Bian et al. [17] used the joints detected from a Kinect camera to define features and detect fall. They used SVM to classify the actions into fall and non-fall.

Yao et al. [122] also used Kinect camera, but they focused on extracting the torso vector. They detected fall based on torso angle. This approach still suffers from not considering velocity of the action.

Merrouche et al. [76] suggested that tracking head only is adequate for fall detection applications. Therefore, they detected head and ground and calculated the distance between head and ground at each frame to detect fall. However, if the ground is bumpy, such as stairs and obstacles, then detecting the floor plane is not straight forward and can be challenging.

Some reserachers, such as [35], fuse different forms of data (such as video and sound) together to get more accurate detection.

A new trend that is getting more attention in recent years is the use of Convolutional Neural Network (CNN) and deep learning to detect fall. Some researchers [32], [36], [64], [65] used different networks such as RNN, faster R-CNN and so on to detect fall. However, the problem with all these approaches is the training dataset. If the training set is not representative or not sufficiently large, then these techniques fail. An example is the illumination condition and viewing direction of the camera. If they are significantly different between training and testing data, then theses techniques fail.

3.3 Proposed Method

Most of the vision based fall detection techniques that have been proposed in the literature works on action level [32], [36], [64], [65], [76], [77]. Which means breaking the captured video into smaller action and detecting fall in short clips. There are few disadvantages with this approach. What is the

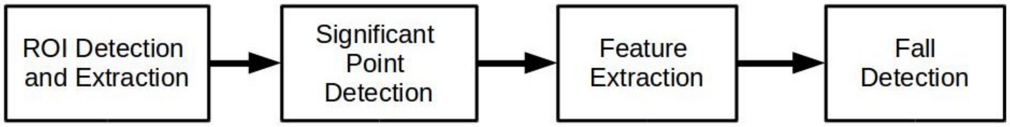


Figure 3.1: General outline of the proposed fall detection system.

strategy to divide the clip into smaller actions? How should one deal with the border in activities? What would happen if the video is broken in the middle of the action? These ambiguity suggests that detecting fall in the frame level can be more desirable. The fall detection system we propose in this Chapter is real time and works at the frame level. It contains different stages as shown in Fig. 3.1.

The first step is extracting a region of interest. Unlike other single camera fall detection techniques [77], [93], [97], [123] and [35], where they used background subtraction only, we introduce a robust background subtraction technique so that the person can be detected and extracted more robustly and accurately.

The second step is detecting two significant points which are needed later to extract the set of features and detect fall accordingly. As [92] and [76] suggested, and our experiments show, fall can be detected by having the head, h, and center of person, c, positions. We chose these points as they are visible and less occluded comparing to other points in legs or arms. The vector formed by these points is called **ch**.

The third step is extracting features based on two detected points and extracted ROI. We used four features in this chapter including:

- The size of **ch**, which will be shown by $||\mathbf{ch}||$ in this chapter.

- Angle of **ch** with respect to horizontal line, and is represented by $\hat{\mathbf{ch}}$.

- Average angular velocity of **ch**, shown by $v_{\mathbf{ch}}$.

- The ratio of bounding box, (r_1, r_2), around the region of interest.

Since we have a combination of different features that explains the posture of the person and velocity of movement, our proposed approach is more robust to false positive as demonstrated by our analytical results.

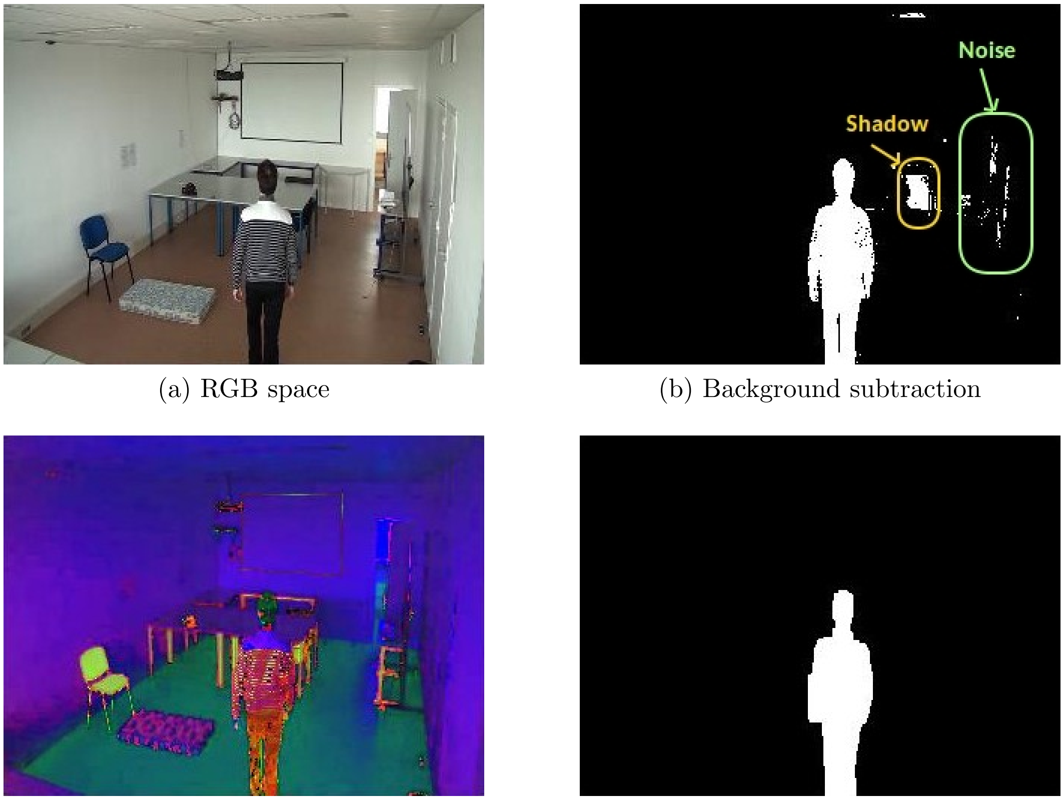
The last step is detecting fall. We used unsupervised classification based on threshold to detect abnormal activities. The abnormal action is falling in this Chapter.

3.3.1 ROI Detection and Extraction

Background Subtraction

Background subtraction or foreground segmentation is an important step for different computer vision applications, such as surveillance. Different techniques have been proposed in the literature. [51] compared some of these techniques. Based on this paper and our experiments, we chose the Gaussian Mixture Model (GMM) method proposed by [103]. This model is quite suitable for a dynamic background, since a mixture of K Gaussian distribution is used to model the temporal histogram of each pixel throughout the video. This technique requires a training phase in order to estimate the background. Note that since it is a pixel based background detection, it is quite fast and suitable for real time applications.

The differences in intensity value of each pixel location (i, j), in each frame k can be due to a moving object or random changes or noise present in the scene. For example, changes in illumination due to shadow or highlights or small movements of random object in the scene due to external forces, like wind, can change the intensity value of each pixel. A set of different Gaussian functions are used to model the intensity value of each pixel. After the training



(a) RGB space

(b) Background subtraction

(c) Chrominance component

(d) Background subtraction

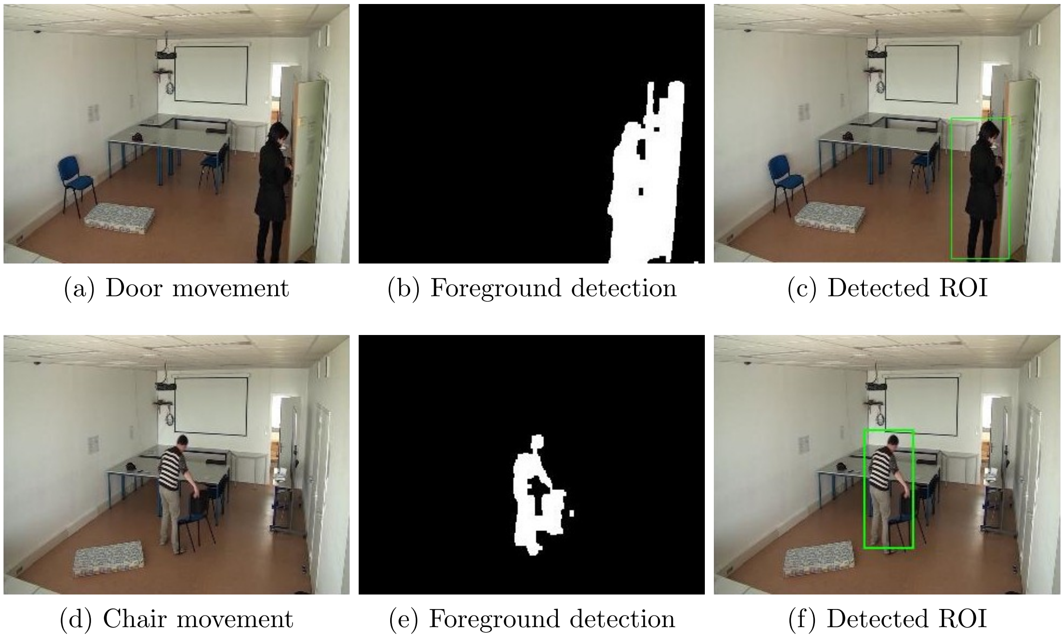
Figure 3.2: Comparison between background subtraction on RGB space and chrominance component of the same frame. The same GMM-based background subtraction is applied on both images.

phase, the intensity value of each pixel is compared with the Gaussian mixtures and based on their probability the algorithm decides if that pixel belongs to a moving object or not.

Removal of Illumination Changes

Although GMM-based background subtraction is suitable for dynamic scenes, it fails if the unwanted changes in the scene are relatively large. For instance the shadow of a moving person is still detected as a part of the foreground. This can be problematic for fall detection applications, where we are only interested in movement of the person.

To address this problem we seperated the color space into luminance and



(a) Door movement (b) Foreground detection (c) Detected ROI

(d) Chair movement (e) Foreground detection (f) Detected ROI

Figure 3.3: (a,d) Original frame of two different actions involving movement of person and other objects. (b,e) Detected moving objects using GMM foreground detection only. Note that door and chair are also detected. (c,f) Extraction of the person only as the ROI. Note the removal of door and chair from the bounding boxes.

chrominance components as suggested in [48]. Some color space like HSV or YC_bC_r give a better estimate of chromaticity and luminance. Background subtraction in chromaticity eliminates the detection of movements due to illumination changes. Fig. 3.2(b) shows the result of applying GMM background subtraction on the original RGB frame. The yellow region is the shadow of the person on the wall and the green area is noise created by changes in illumination conditions due to shadow. Fig. 3.2(d) shows that by applying GMM background subtraction on chromaticity component only, we can eliminate these unwanted regions and noise due to illumination changes.

Unwanted Objects Removal

Another problem is the movement of random objects, which can be smaller or bigger than the person in the scene. Fig. 3.3(a) shows the movement of the person and the door in the room, and Fig. 3.3(d) shows the person carrying a chair with himself. We would like to get rid of these random motions as well and only focus on the person. In other words we are interested in detection and extraction the Region of Interest only, rather than foreground. [97] proposed removing unwanted regions by keeping the region with the biggest area. However, this technique will fail if some bigger objects are moving in the room such as, the door in Fig. 3.3(a) or a piece of big furniture like a sofa. Therefore, we proposed a new ROI detection technique based on the area of region and penalty function. The proposed method fuses background subtraction and person detection together and removes errors and noise based on a penalty function. Note that person detection and tracking techniques, such as those proposed in [27], [28], mainly work on people with upright postures and fail if the person is sitting or lying down. Hence, it is not a suitable approach for fall detection by itself.

Eq. 3.1 summarizes our novel ROI detection technique, where PD is the region detected by the person detector and FD is the region detected by the foreground detector. w_1 and w_2 are their corresponding weight respectively, where $0 < w_1 < 1$, $0 < w_2 < 1$ and $w_1 + w_2 = 1$. We call w_1PD, *effective person* and w_2FD *effective foreground.*

$$ROI = (w_1PD + w_2FD) - \epsilon \tag{3.1}$$

Eqs. 3.2 and 3.3 explain how to calculate the weights. However, note that if the algorithm detects no movement in the scene and only detects a person then w_2 will set to zero and w_1 will be one. The same will happen if it only detects movement and no person. In this case w_1 will set to zero and w_2 will be one.

$$w_1 = \frac{Area(PD)}{Area(FD) + Area(PD)} \tag{3.2}$$

$$w_2 = \frac{Area(FD)}{Area(FD) + Area(PD)} \tag{3.3}$$

In Eq. 3.1, ϵ denotes the penalty function which is explained in Eq. 3.4. The penalty function removes any small regions that are detected due to noise or unwanted movements of small objects. It mainly suggests that in each frame, k, any detected regions with areas smaller than half of the ROI area detected in the previous frame should be removed. R is any closed area that was detected by the proposed technique.

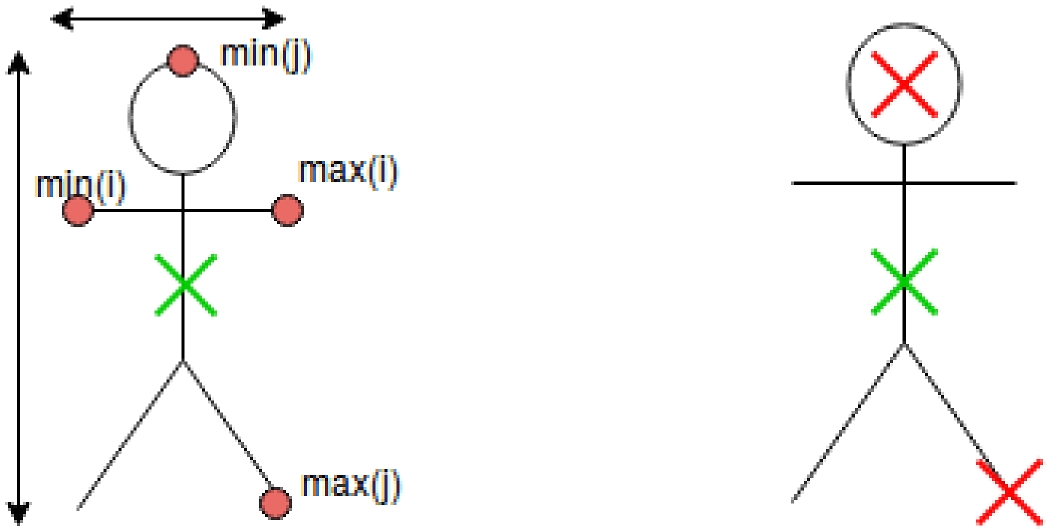
$$\epsilon = (\sum_{i \in R} i < \frac{area_{k-1}}{2}) \tag{3.4}$$

We used HOG features to extract PD. This detects people in the upright position as proposed in [27] and the formulation is given in Eq. 3.5, . In other words gradients in x and y directions is calculated for each pixel in each cell c. This is called g_x and g_y. Then based on orientation of gradient, $\theta = atan(\frac{gy}{gx})$, a histogram with 9 bins is formed for each cell. The magnitude of gradients for each pixel is then located in the corresponding bin.

$$HOG = \sum_{i=1}^{9} (\sqrt{(g_x)^2 + (g_y)^2})_i \tag{3.5}$$

In order to extract FD, we used a GMM-based foreground detector. This technique is proposed in [103]. Each pixel in the image is presented by a mixture of k Gaussian distribution as shown in Eq. 3.6. The most probable Gaussian distributions belong to the background and the least probable ones represent the foreground.

$$P(\mathbf{X}_t) = \sum_{i=1}^{k} \alpha_i \eta(\mathbf{X}_t \mu_i, \sigma_i) \tag{3.6}$$



(a) Feret diameter

(b) Optimization

Figure 3.4: Feret diameter intuition and ambiguity in head detection. Green cross is the center of person and the red crosses are potential locations for head.

Here α_i is the weight and μ_i and σ_i are the mean and standard deviation of the i^{th} Gaussian distribution.

Fig. 3.3(c,f) shows the bounding box around an extracted ROI. One can see that the ROI detection only chooses the person as ROI and not the door or the chair. This would not be possible if we only applied the foreground detection technique as Fig. 3.3(b,e) shows that door and chair are also detected by foreground detection.

3.3.2 Significant point detection

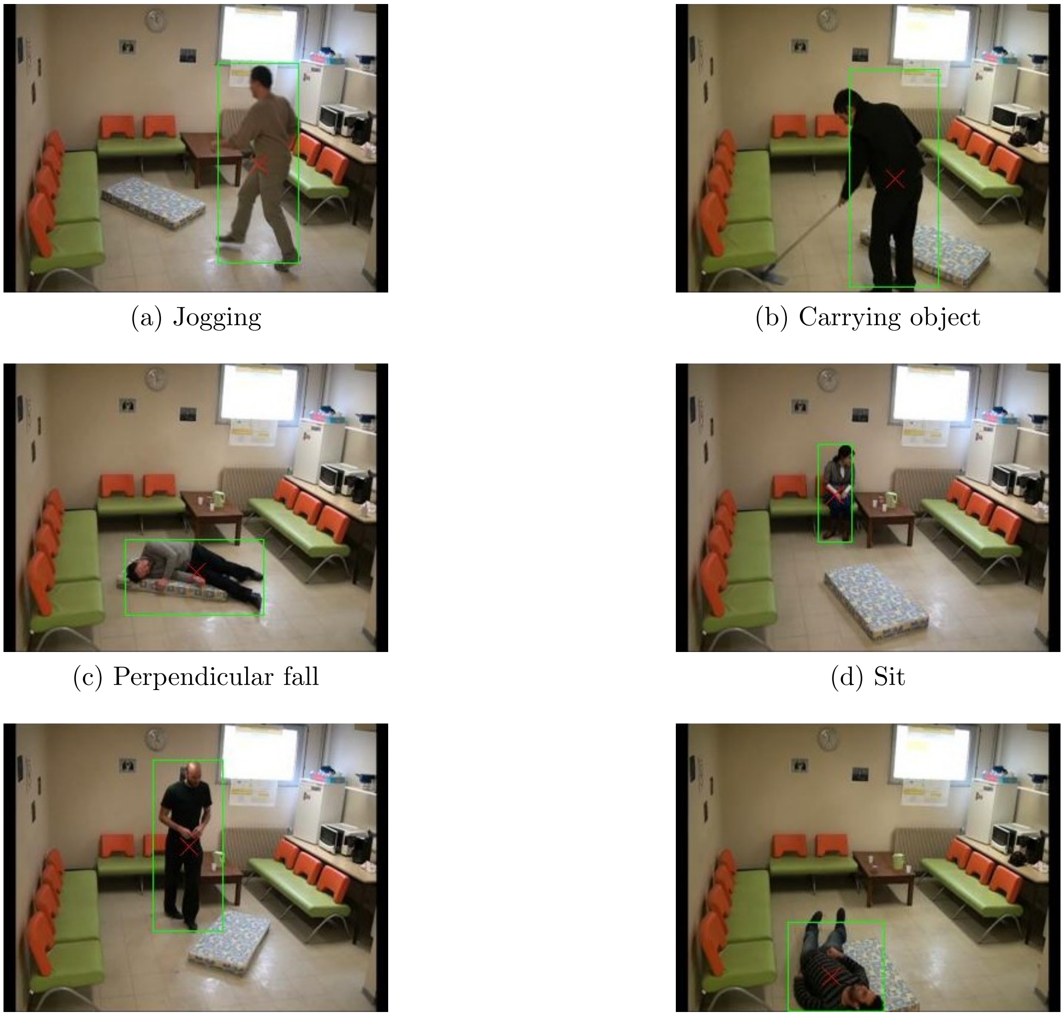
In order to detect fall we need to extract a number of significant features. People either used the posture features extracted from the bounding regions, such as [77], [93], [123], or they analyzed the trajectories of some important points or vectors like in [24], [76], [92], [97] . In this chapter we used bounding box information as well as trajectory and velocity information of the significant vector. To form this vector we used head h and center of person c position as two significant points. The reason for having different sets of features is to be more robust in detecting fall. For example, having the bounding box information can help in detecting falls which occur along the viewing direction of the camera. By using the velocity information we can distinguish falling down from sitting down or lying down. Note that tracking one keypoint only, like in [92], will create error in 2D space as we loose one degree of freedom.

Center of person detection

The key point c represents pelvis joint which is the intersection of maximum and minimum feret diameter of the extracted ROI and is given by Eq. 3.7, where each pixel in the extracted ROI has the coordinate of (i, j). Also, Fig. 3.4(a) shows the intuition behind feret diameter and center of person calculation. The small circles correspond to the maximum and minimum i and j location of the detected ROI in the xy coordinate system and the green cross shows the calculated center of person which roughly represents the pelvis joint. Note that based on position and orientation of the detected ROI, the location of the maximum and minimum point and the detected center of person will change.

$$c = (\frac{max(i) - min(i)}{2}, \frac{max(j) - min(j)}{2}) \tag{3.7}$$

Fig. 3.5 shows the detected center of person for different people with different actions. Observe that the position is quite accurate even when the person is carrying another object like a broom as Fig. 3.5(b) shows. However, if the person is falling along the viewing direction of the camera, z axis, then c does not exactly represent the pelvis. Fig. 3.5(f) shows this situation. Later we will show that it does not affect the accuracy of our fall detection technique and we can still detect fall in these cases, using the ratio of the bounding box and the velocity of the action.



(a) Jogging

(b) Carrying object

(c) Perpendicular fall

(d) Sit

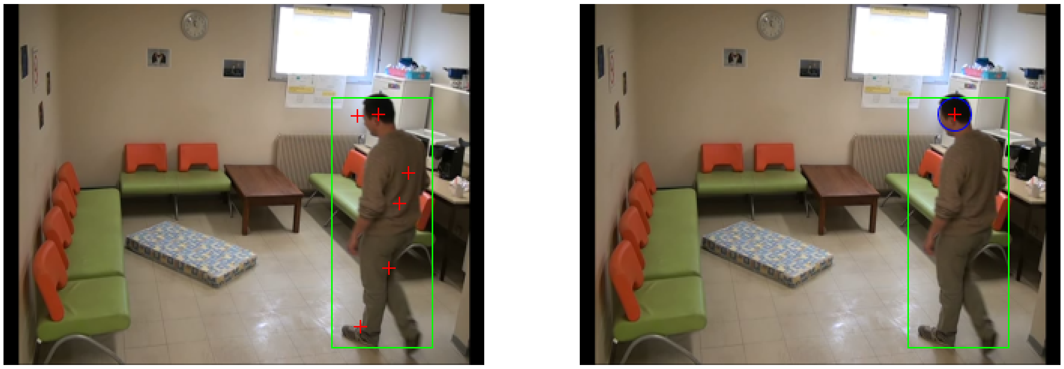
(e) Standing

(f) Z-fall

Figure 3.5: The extracted center of person is shown by the red cross

Head detection

The second key point is head position. Detecting head is a crucial step for surveillance applications as [92] suggested. Researchers have used different techniques to estimate the head position. [71] proposed a head detection approach which can work on sideview and backview of the head as well. However, this approach is very time consuming and not suitable for a real time application. Some reserachers suggested that the human head can be approximated by circle or ellipse [66], [91], and they used different feature extractor to detect imperfect circles or ellipse. Here we used circular hough transformation on the detected ROI to find circular curves and potential head positions. As [29] proposed, hough transformation works based on a voting system and therefore it can detect several imperfect circles with the specific radius. The center of these circles can be the potential positions for the head. Fig. 3.6(a) shows an example of the potential head positions marked by red crosses. In order to find the actual head position, first we need to clean the set of centres found and remove the points which are: 1) outside the ROI, or 2) inside the ROI, but more than half of the area of the circle formed is outside the ROI. Now, we have a set of candidate points which we call h'. We used an iterative two step optimization to find the best head position. The optimization only keeps centres with largest Euclidean distance from c, if its reflection point with respect to c is inside the ROI. An example of potential head position and final head position is shown in Fig. 3.6(a) and Fig. 3.6(b) respectively. Eqs. 3.8 and 3.9 summarize the optimization process. The intuition behind the second step of optimization is shown in Fig. 3.4(b). The two red crosses are two candidate head points h'_1 and h'_2 and the green cross is the center of person c. It can be seen that the Euclidean distance between h'_2 and c is greater than that of h'_1 and c. However, the reflection of h'_2 with respect to c will be out of ROI;



(a) Candidate head positions (b) Final head position

Figure 3.6: The steps and result of the head detection approach.

while this is not the case for h'_1.

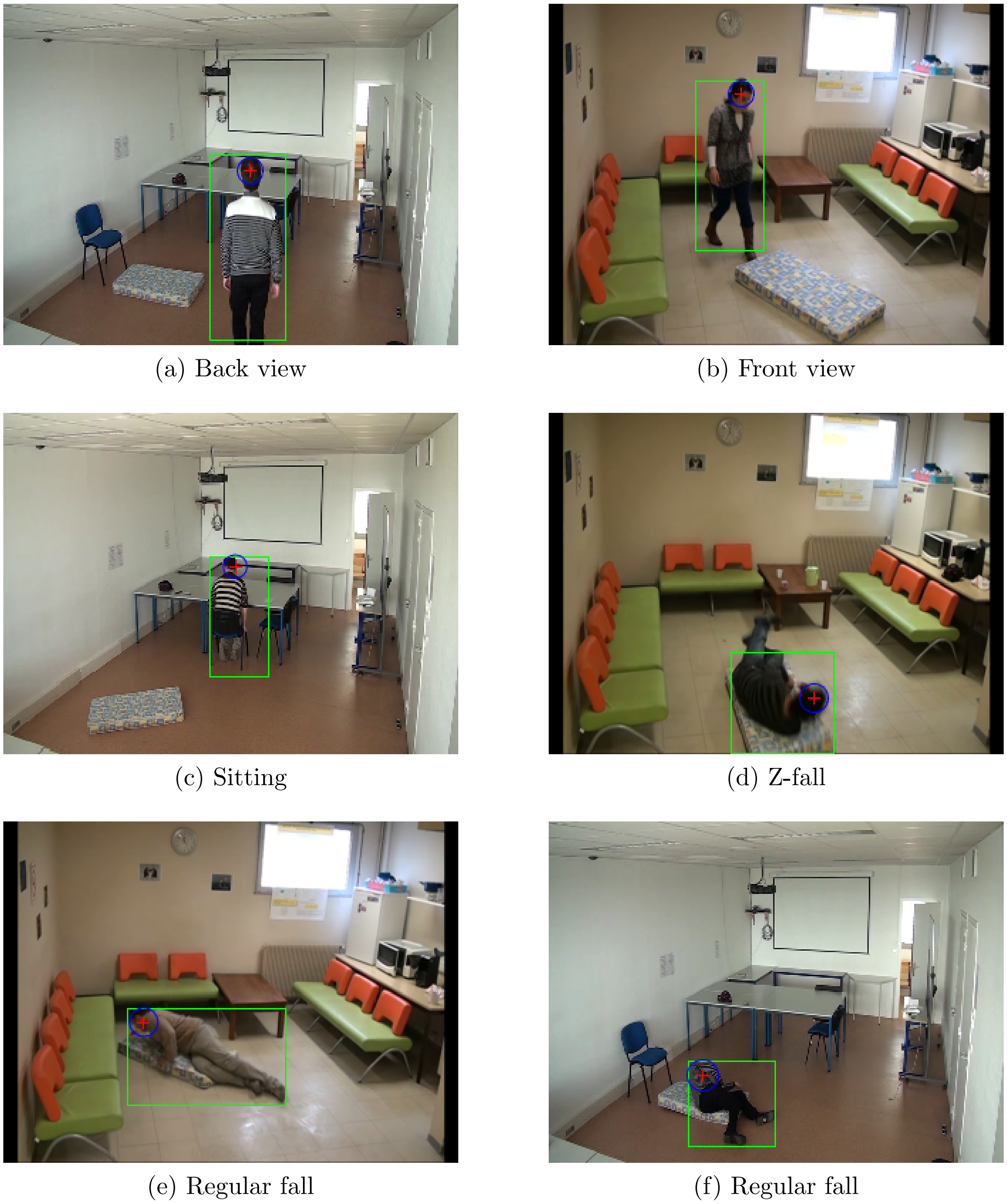
$$h = \max_{h'}(\sqrt{(j'_h - j_c)^2 + (i'_h - i_c)^2}) \tag{3.8}$$

$$((2i_c - i_h), (2j_c - j_h)) \in ROI \tag{3.9}$$

Fig. 3.7 shows more examples of the head detection step applied on different people with different postures. This figure shows that the proposed head detection algorithm is quite accurate and robust and can be useful for different surveillance applications. It is very fast and it can work on any position and direction of the head even for a low resolution video. We have tested our algorithm on two different video resolutions. On the average it takes $0.017 sec/frame$ for a lower resolution video of 320 × 240 pixels to find the head position. However for a higher resolution video of 850 × 480 pixels, it takes $0.048 sec/frame$, which is still real time. Note that in order to optimize the time performance without affecting the accuracy, the video can be reduced to a lower resolution.

3.3.3 Feature extraction

The next step is defining some distinct features that can capture the characteristics of falling down. As [83] showed, falling down is a fast action, and


(a) Back view

(b) Front view

(c) Sitting

(d) Z-fall

(e) Regular fall

(f) Regular fall

Figure 3.7: The result of the proposed head detection approach for different postures and people.

its critical phase is between 300 to 500 milliseconds. Note that even for cases when a person is gradually falling down, like from a bending/sitting position or when holding onto something and then falling, there still exist a fast and critical phase. Otherwise, it is impossible to distinguish falling down from lying down for instance. In this chapter we combined posture and velocity information to detect fall. We used four different features. The first feature is the ratio of bounding box around ROI, which is given in Eq. 3.10. Where r_2 is the height and r_1 is the width of the bounding box.

$$f_1 = \frac{r_2}{r_1} \quad \text{and} \quad f_1 \in [0, 8] \tag{3.10}$$

Almost all the previous vision based fall detectors use bounding box information, such as area or ratio, as a posture feature. It is due to the fact that the bounding box information is simple, fast and reliable.

The ratio of the height and width of the bounding box is a real number between 0 and 8. Ratios greater than 1 mostly represent an upright posture. An exception can be falling down in the viewing direction of the camera (z-fall) where f_1 can be greater than or equal to 1, as can be seen in Fig. 3.5(f) and Fig. 3.7(d).

To address this issue we use $||\mathbf{ch}||$ which is the vector formed by center of person, c and head, h as explained in Section 3.3.2. The intuition is that due to perspective projection the length of the person would not change in standing and perpendicular lying position, assuming the distance of the person to the camera does not change. However, for all the other lying position with the same distance to the camera, the length of the person would be shorter. Later we will explain how to handle different distances to the camera. An example of perpendicular fall is given in Fig. 3.5(c). However, we should normalize this feature so that it can work for people with different heights. Eq. 3.11 is used to get the second feature, where $h/2$ is half of the bounding box generated by the person detector technique, explained in Section 3.3.1. Since $h/2$ is updated whenever the person detector detects an upright person, we do not need to worry about the distance of the person to the camera.

$$f_2 = \frac{||\mathbf{ch}||}{h/2} \quad \text{and} \quad \in [0, 1] \tag{3.11}$$

These two features are not sufficient to capture posture information for different actions, such as bending or sitting down; therefore, we use the angle of significant vector, **ch**, w.r.t the horizontal line. Since the direction of the action is not important for us, this angle is between $[0, \frac{\pi}{2}]$. This feature is presented in Eq. 3.12, where (x_c, y_c) and (x_h, y_h) show the x and y coordinates of the center of person and head respectively.

$$f_3 = atan(\frac{|y_c - y_h|}{|x_c - x_h|}) \quad \text{and} \quad f_3 \in [0, \frac{\pi}{2}] \tag{3.12}$$

The last feature captures the velocity of the action. The velocity is crucial to distinguish between intentional lying and falling. We used the average angular velocity of **ch** between each five frames.This corresponds to 200 ms for a 24 frames per second video. This feature is shown in Eq. 3.13.

$$f_4 = |f_{3_k} - f_{3_{k+5}}| \quad \text{and} \quad f_4 \in [0, \frac{\pi}{400}] \tag{3.13}$$

3.3.4 Fall detection

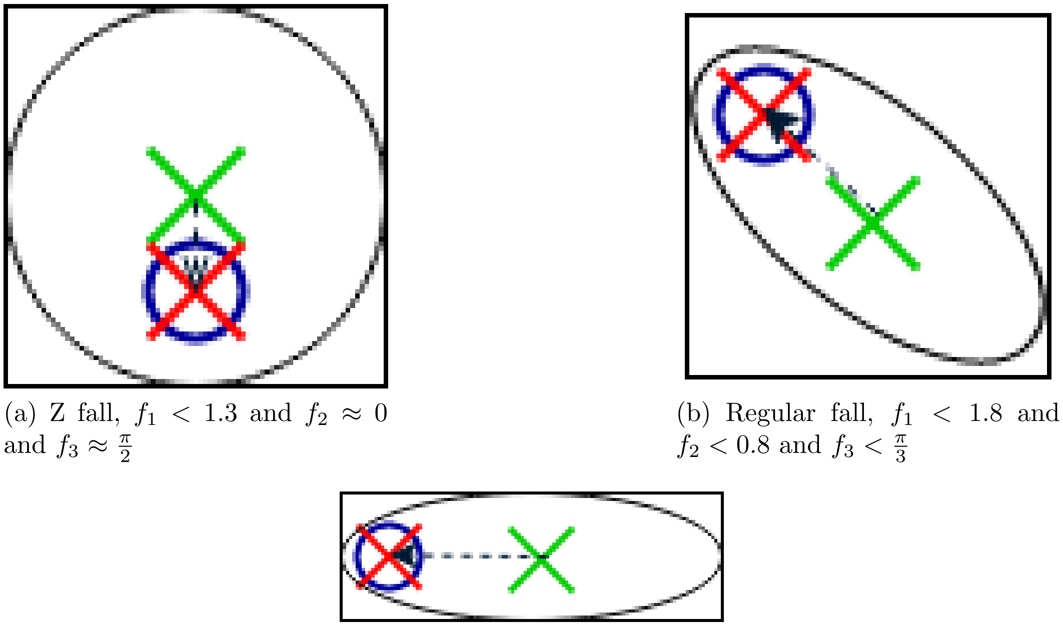
The last step of the proposed method is detecting fall based on the extracted features. It would be quite challenging to use supervised learning at the frame level detection techniques because of the ambiguity in the border of different action. More specifically one can ask what is the starting point of the fall. In this chapter we label each frame as *non-fall* or *potential fall* by thresholding on the features. To find the best threshold for each feature we used grid search.

If a frame is classified as *potential fall* for five consecutive frames, which is equivalent for 200 ms, then a fall is detected. The reason for choosing five frames is to detect falls that happen slower, by holding onto something for instance, as well as detecting faster falls. Fig. 3.8 shows different scenarios for a fall event with their corresponding feature values. Note that there can be different orientations of the body with the same feature value, like head down in regular fall, which we did not show here. The big rectangle or square box represents the bounding box around the ROI. The ROI is shown by the big circle in Fig. 3.8(a) and big ellipse in Fig. 3.8(c,d). The green cross represents center of person and the red cross represents the head position. The summarized conditions for fall is shown in Eq. 3.14. Note that angular velocity is an important feature to distinguish fall from other actions.

$$fall = (f_1 < 1.8) \;\; and \;\; ((f_2 < 0.8) \;\; or \;\; (f_3 < \frac{\pi}{3})) \;\; and \;\; (f_4 > 15) \tag{3.14}$$

3.4 Details on the Dataset

We chose Le2i [20] to test our algorithm. The reason is the vast selection of actions actors combinations. The dataset contains 191 video sequences where 143 videos show falling action and 48 videos does not have fall. Only one actor is presented in each video, and there are nine different actors in total with different body features and different outfits. Different everyday activity, such as walking, bending, sleeping, sitting down, standing up, putting on clothes, talking over the phone and so on are included in each video. The actions happen at four different indoor environments including lecture room, coffee room, home and office, where the viewing directions of the camera and the lighting conditions of the rooms are different. The videos are captured at 25 frames per second and the resolution is down sampled to 320 × 240.



(a) Z fall, $f_1 < 1.3$ and $f_2 \approx 0$ and $f_3 \approx \frac{\pi}{2}$

(b) Regular fall, $f_1 < 1.8$ and $f_2 < 0.8$ and $f_3 < \frac{\pi}{3}$

(c) Perpendicular fall, $f_1 < 1$ and $f_2 \approx 1$ and $f_3 \approx 0$

Figure 3.8: Different scenarios for fall and their corresponding features. The black circle or elipse is the ROI, the blue circle is head, the green cross is c and the red cross is h.

3.5 Experiments and Results

We have calculated accuracy, recall and specificity of the proposed method which are given below.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{3.15}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.16}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3.17}$$

TP and TN coressponds to *True Positive* and *True Negative* respectively and is the number of frames correctly classified as fall or not-fall. FP and FN are *False Positive* and *False Negative* and is the number of frames that are labeled incorrectly as fall or not-fall. Note that the classification happens at the frame level, and fall duration is only 14 frames for most of the cases [20].

Table 3.1: Acurracy, specificity and recall of the proposed fall detection system.

Room	Accuracy	Recall	Specificity
Lecture room	99.45%	100%	99.12%
Coffee room	98.9%	89.8%	99.75%
Home	99.42%	95.27%	99.83%
Office	99.7%	90.58%	99.96%

Thus, reporting accuracy only is not sufficient to evaluate any fall detection technique. However, most of the papers in the literature only reported accuracy which is not a fair indication of how effective a fall detection technique is. Table 3.1 shows Accuracy, Recall and Specificity of our proposed method, for each indoor location that was available in the dataset. In order to have a fair comparison, we reported for each room separately. The lower recall values for Coffee room and Office is because there is a significant amount of Z-fall for these location.

Fig. 3.9 shows the result of the proposed method for different fall scenarios and some of the failed cases. Fig. 3.10 shows the result of the proposed method on other types of action. As Fig. 3.9(d,i,k) shows, there might be cases where the detected head position is wrong. It can happen for various reasons like the head being not completely visible. However, in such cases we can still detect fall or no fall action accurately due to our robust set of features. Also, when the person is in lying position the enclosed bounding box might be bigger than expected, such as in Fig. 3.9(a,b). This happens because the mat is not fixed to the ground and it moves when the person fall but again our robust features can detect fall accurately. Note that despite our attempt to detect Z-fall, there are situations that our algorithm cannot detect. In fact, all cases of *False Negative* belongs to Z-fall. We can detect Perpendicular-fall and Regular-fall with 100% recall. The proposed method can correctly label not-fall action for almost all activities including even bending and squatting.

Table 3.2 compare the results of the proposed method with two other tech-

Table 3.2: Comparison of the proposed fall detection system with other techniques.

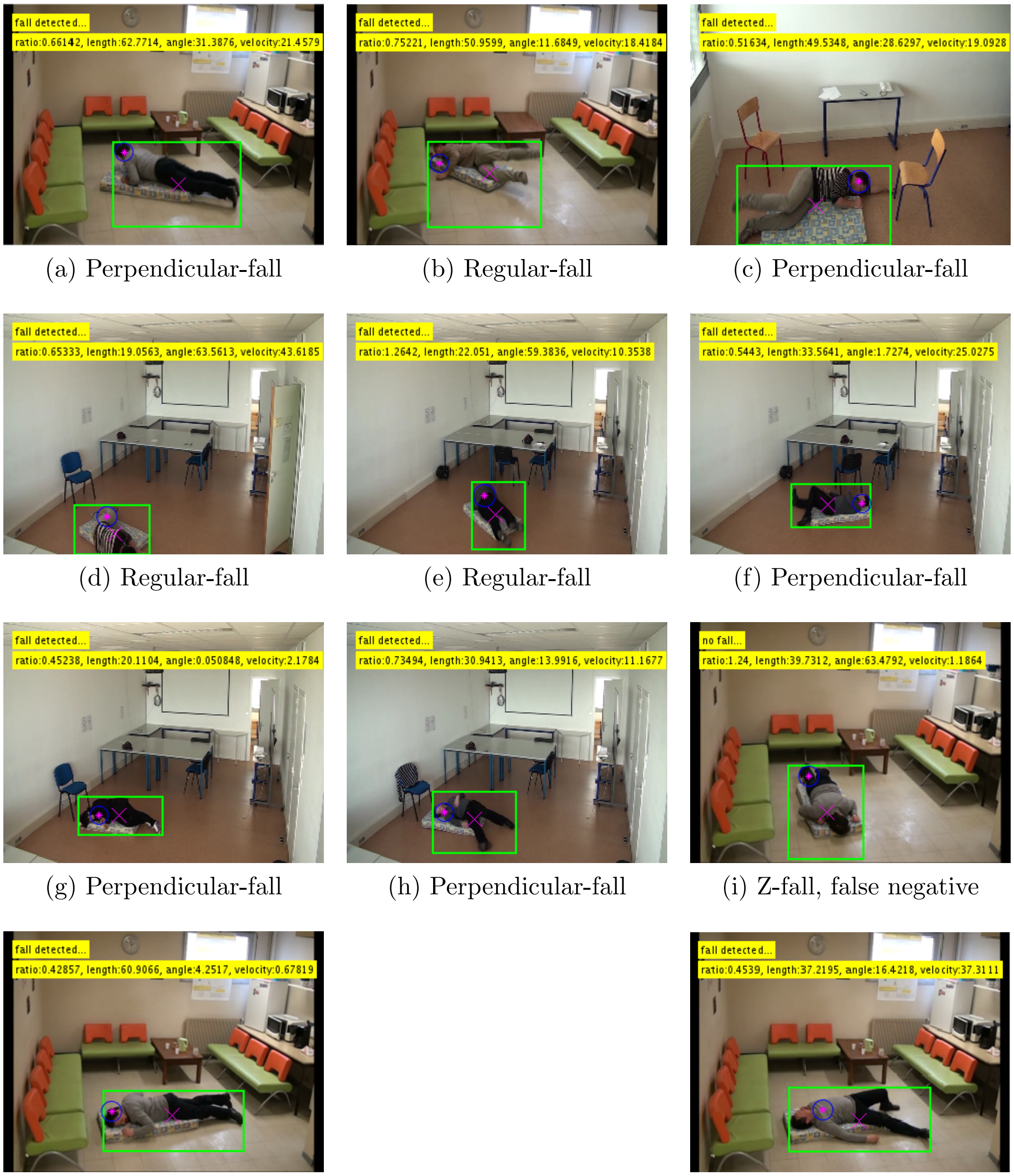
	Proposed Method	Method proposed by [20]	Method proposed by [97]
Accuracy	99.36%	99.54%	99.61%
Recall	93.91%	84.84%	NA
Specificity	99.67%	90.32%	NA

niques that applied on the same dataset. It is important to note that there have not been many works that reported percision and recall. Also they did not test their proposed method on different room setups. Table 3.2 shows the average accuracy, recall and percision for the proposed method.

3.6 Conclusion and Future Work

In this chapter, we proposed a real-time automatic fall detection system based on motion deformation and velocity of the action. We found that fall detection at the frame level is more desirable than action level due to ambiguity in the starting point of the action and how to break a long video into smaller actions. The proposed method detects the region of interest which is the person through a novel and robust ROI detection technique. The method also detects two significant points, head and center of person. We then use the angle and size of the vector between head and center of person as well as its angular velocity to detect fall based on a set of thresholds which is obtained by grid search. The results show that the proposed system is quite accurate and very robust with over 99% accuracy, and over 93% recall and over 99% specificity on the average. The most frequent false negative happens for Z-fall.

In future we would like to use depth camera to address the issues in Z-fall. Also, we would like to use an outlier detection approach, such as One Class SVM (OCSVM) to detect fall as an outlier action.



fall detected...
ratio:0.66142, length:62.7714, angle:31.3876, velocity:21.4579

(a) Perpendicular-fall

fall detected...
ratio:0.75221, length:50.9599, angle:11.6849, velocity:18.4184

(b) Regular-fall

fall detected...
ratio:0.51634, length:49.5348, angle:28.6297, velocity:19.0928

(c) Perpendicular-fall

fall detected...
ratio:0.65333, length:19.0563, angle:63.5613, velocity:43.6185

(d) Regular-fall

fall detected...
ratio:1.2642, length:22.051, angle:59.3836, velocity:10.3538

(e) Regular-fall

fall detected...
ratio:0.5443, length:33.5641, angle:1.7274, velocity:25.0275

(f) Perpendicular-fall

fall detected...
ratio:0.45238, length:20.1104, angle:0.050848, velocity:2.1784

(g) Perpendicular-fall

fall detected...
ratio:0.73494, length:30.9413, angle:13.9916, velocity:11.1677

(h) Perpendicular-fall

no fall...
ratio:1.24, length:39.7312, angle:63.4792, velocity:1.1864

(i) Z-fall, false negative

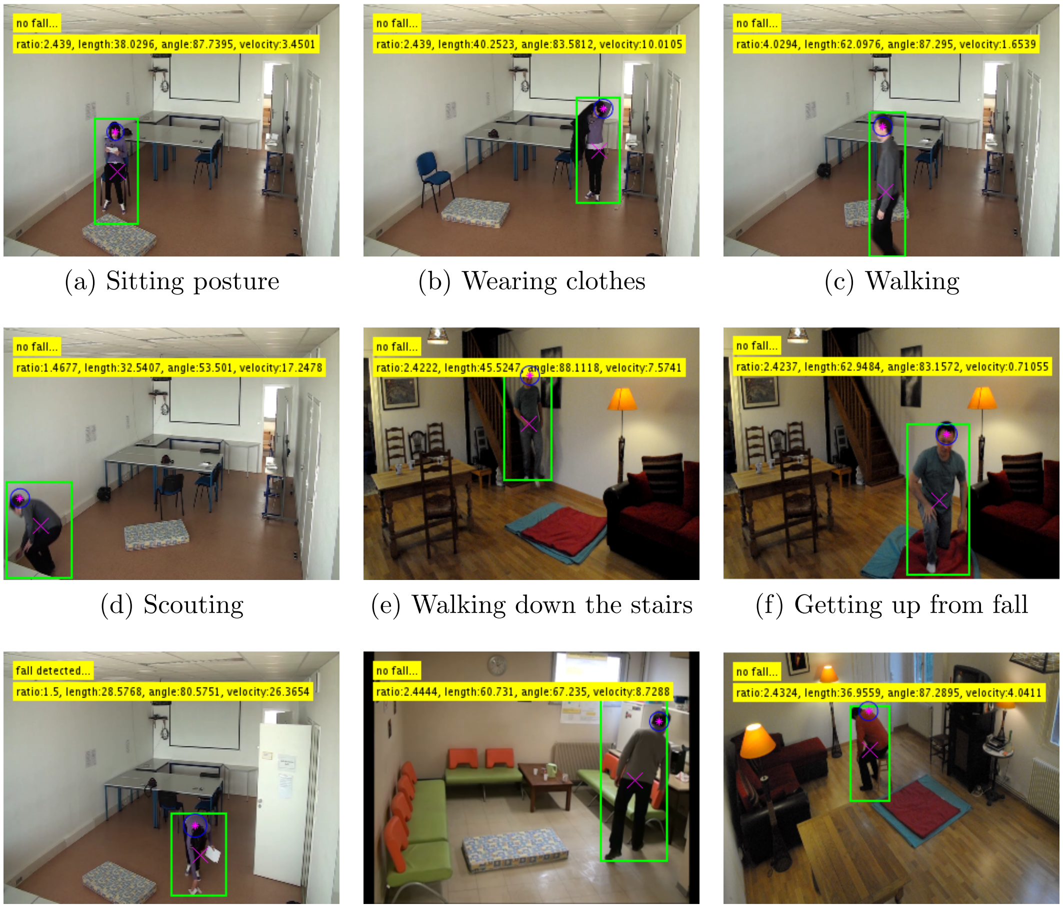
fall detected...
ratio:0.42857, length:60.9066, angle:4.2517, velocity:0.67819

(j) Perpendicular-fall

fall detected...
ratio:0.4539, length:37.2195, angle:16.4218, velocity:37.3111

(k) Perpendicular-fall

Figure 3.9: The result of proposed detection system on fall actions.


no fall...
ratio:2.439, length:38.0296, angle:87.7395, velocity:3.4501
no fall...
ratio:2.439, length:40.2523, angle:83.5812, velocity:10.0105
no fall...
ratio:4.0294, length:62.0976, angle:87.295, velocity:1.6539
no fall...
ratio:1.4677, length:32.5407, angle:53.501, velocity:17.2478
no fall...
ratio:2.4222, length:45.5247, angle:88.1118, velocity:7.5741
no fall...
ratio:2.4237, length:62.9484, angle:83.1572, velocity:0.71055
fall detected...
ratio:1.5, length:28.5768, angle:80.5751, velocity:26.3654
no fall...
ratio:2.4444, length:60.731, angle:67.235, velocity:8.7288
no fall...
ratio:2.4324, length:36.9559, angle:87.2895, velocity:4.0411

(a) Sitting posture

(b) Wearing clothes

(c) Walking

(d) Scouting

(e) Walking down the stairs

(f) Getting up from fall

(g) Bending, false positive

(h) Bending

(i) Sitting down

Figure 3.10: The result of proposed detection system on other actions.

Chapter 4

Evaluation of 3D model segmentation techniques based on animal anatomy

One of the most challenging and important problems in computer graphics, which has recently attracted many researchers' attention, is 3D model decomposition or mesh segmentation. The problem is segmenting a mesh or volume into components based on some geometric or semantic criteria. Mesh segmentation has several applications in modeling, texture mapping, animation, compression, simplification and so on.

There are two different approaches to 3D mesh segmentation [8]. It can be done either geometrically or semantically. In geometric mesh segmentation the mesh is segmented into a number of visually and anatomically meaningless patches based on some surface properties, such as curvature, distance to a fitting plane or surface normal. However, in a semantic approach, the mesh is segmented into visually meaningful parts that are related to relevant features of the shape. For instance one can segment the 3D model of a horse into head, body, legs and tail. The semantic approach attracts more attention because of its strength in different applications, such as animation, classification, shape retrieval and skeleton extraction. The main challenge with this approach is the lack of a consistent evaluation criteria to compare different techniques, since each technique has been developed for a specific application. Therefore, the comparison of different segmentation techniques should be based on the application. Another problem is the definition of meaningful patches for 3D models. For example, one can divide an animal body, such as a horse, into coarse segments, head, body, legs and tail. While others may segment it based on more detailed animal anatomy; thus, the quantitative comparison can be challenging.

Semantically oriented segmentation of quadrupeds is highly relevant for object retrieval and animation. However, because of the abovementioned reasons there is no obvious metric to evaluate the accuracy and quality of segmentation results. Several previous works have been proposed for segmentation evaluation [15], [22]. They rely on databases of human segmentation of 3D models. However, we argue that arbitrary human observers do not necessarily provide a correct semantic ground-truth to evaluate the segmentation. In this chapter the similarity between segmentation results and an animal anatomy is considered as the evaluation metric.

In this work we also extend an anatomy preserving segmentation technique. The underlying idea is using the skeleton of a 3D model to guide the decomposition. The skeleton of a 3D model is a compact graph-like abstraction derived from the centerlines of the original model [100]. The next step is mapping surface points onto skeleton branches. Finally, each set of surface points that map onto the same skeleton branch is considered as one patch. This approach uses both global shape and local features, helping enhance perceptual quality, which is required in animation and game type applications. In prior work, [100] used only one anatomy model (a horse) as the ground truth and evaluated the segmentation results only for the horse model. This is not adequate for a general finding. In this chapter, we segment other 3D animal models and compare the results with animal anatomy to evaluate various algorithms.

As mentioned above, ground truth for evaluation of segmentation results has been typically obtained solely through human input [15], [22], [72] in the computer graphics and vision communities. However, this approach is purely subjective and could vary significantly from one user group to another. The importance of accurate and consistent segmentation has been pointed out by researchers in various application communities. For example, why defining ground truth for tables [121] is difficult has been discussed in [44]. Defining ground truth for various fields in medicine has been extensively researched, e.g., [89]. Validation of the segmentation and expert definitions of ground truth has also been carefully measured [118]. By contrast, the expectations of the quality of ground truth in vision and graphics research do not seem to be that high. For example, as [72] showed, closed objects may grouped together and identified as one object or separate objects. Despite issues like this, the different segmentations are considered to be "highly consistent." It appears that application domains, and what level of accuracy and consistency are necessary in them, have been largely ignored in vision and graphics research. Thus, we propose using biologically defined anatomical models as a more reliable approach to ground truth definition and subsequent evaluation for a class of objects in 3D segmentation.

The remainder of this chapter is organized as follows: Section 4.1 discusses some of the semantic segmentation techniques. Section 4.2 explains our skeleton based segmentation approach. In Section 4.3 the evaluation criteria and the proposed anatomical ground-truth are presented. Section 4.4 compares the results of skeleton based segmentation with the most efficient state of the art methods. Concluding remarks are given in Section 4.5.

4.1 Related work

Semantically oriented segmentation techniques try to generate the segmentation patches such that a cost function is minimized based on a given criterion. The main difference among various algorithms in this category is the cost function and the criterion used. In this section we will explain some of these techniques in greater detail.

The authors in [53] proposed a hierarchical decomposition technique using fuzzy clustering. This algorithm is based on a hierarchical tree. Each node in the tree is associated with the mesh of a particular patch and the root is associated with the whole input object. The higher level nodes correspond to coarser patches while leaves and lower level nodes correspond to finer patches. The algorithm determines a suitable number (k) of patches at each node, and then computes a k-way segmentation of this node. First, the algorithm finds the meaningful components along with the boundaries between the components that are considered to be fuzzy. The next step is finding the exact boundaries in the fuzzy areas which preserve the features of the object. However, the approach relaxes the condition that every face should belong to exactly one patch and allows fuzzy membership, which means that each face has probabilities associated to belonging to different patches. This probability is based on geodesic and angular distances between all pairs of the faces. Another hierarchical method has been proposed in [52]. This algorithm is also based on a hierarchical tree and proceeds from coarse to fine scale. The main advantage of this technique is that it is insensitive to pose and proportions. The approach first transforms the mesh vertices into a pose invariant space, then robustly extracts the feature points, and finally extracts the core component of the mesh. [7] proposed an algorithm based on the fitting primitives. Initially, each triangle of a triangular mesh corresponds to a single cluster. All the pairs of adjacent clusters are considered and compared at every iteration, and the one that can be best approximated by one of the primitives forms a new single cluster. The primitives are planes, spheres and cylinders; and, an L2 metric is used to compare each combination of the cluster with one of these primitives.

The Shape Diameter Function (SDF) is another 3D segmentation approach which has been proposed in [99]. The SDF is defined as the diameter of the object in the neighborhood of each point on its surface. Given a point on the surface mesh a set of rays is sent inside a cone centered around its inward-normal direction (the opposite direction of its normal) to the other side of the mesh. The value of the SDF at the point is defined as the weighted average of all the lengths of rays that fall within one standard deviation from the median of all lengths. [2] presented a segmentation technique based on random cuts. The idea is to generate a random set of mesh segmentations, and then measure how often each edge of the mesh lies on a segmentation boundary in the randomized set. An interactive approach based on random walk has been proposed in [62]. The user chooses some faces as seeds, then a probability is assigned to each of the three edges of non-seed faces. This value determines whether or not a random walker moves across a particular edge to the corresponding face. A face belonging to the region is grouped with the seed X if a random walker starting at that face has a higher probability of reaching the seed X than any other seeds.

4.2 Segmentation based on Skeletonization

Since the human eyes are very sensitive to changes along the boundary of an object it is logical to use the structural shape for simplification and segmentation purposes. Since the projected 2D contour of a 3D model can vary significantly depending on a change of view, it is more effective to use the model skeleton in order to guide segmentation. The skeleton of a 3D model is a compact graph-like abstraction derived from the centerlines of the original model [100].

It is very important to extract the skeleton as accurately as possible. For segmentation purposes a unit width skeleton, such as the Valance Normalized Spatial Median (VNSM) [116] is needed. However, these skeletons are very prone to local and global noise and may contain some unwanted branches. The Scale Space and Gaussian filters can be used to remove the noise and smooth the skeleton. The unwanted branches should be removed considering the length and topological position of a branch [100].

After extracting the smooth and noise-free skeleton, the next step is to decompose the model into different segments according to the following steps.

- Decompose the skeleton at junction points. Junction points are the points that are connected to more than one skeleton branch.
- Map surface nodes to skeleton branches. This is based on the L2 distance between surface nodes and skeleton vertices. In other words, to map a point on the surface to a skeleton branch node, it should be close to the skeleton branch nodes, with skeleton-to-surface normal vector pointing in a direction similar to the surface normal vectors. Also, the neighboring nodes should have the same label if the curvature is close. These distances are computed for each surface point or a subset of surface points against all skeleton vertices.
- The last step is labelling the neighbourhood nodes. The previous step just labeled some selected sample points. The watershed method is used to flood the labels from samples.

4.3 Evaluation

The main challenge for semantically oriented segmentation is defining the evaluation criteria, which is related to the application. However, for animation purposes and generating natural movements of a model, a more meaningful segmentation for animals is based on anatomy. Shi et al. [100] used a metric function distance for evaluation purposes. The error is then the Euclidean distance between registered pair of cuts plus the length of non-paired cuts. However, they just used one anatomy model, that of a horse, as the ground truth and evaluated the segmentation results only for the horse model. This limited evaluation is not adequate.

For evaluation purposes we used the same method as Shi et al. [100]. First, we registered different cuts to the corresponding ground truth ones; then calculated the Euclidean distance for each cut plus the length of non-paired cuts. Smaller errors imply better automatic segmentation compared to the ground truth. Since the ground truth is generated manually based on animal anatomies, different experts may specify slightly different ground truths. However, this does not affect the comparison results very much, since the ground truth is not solely based on human perception but anatomical information is also considered in the process.

We use sheep [75], cow [75], dog [74] and giraffe [1] anatomies. There are several free 3D models of these animals available on public databases [22], [101]. These anatomy models are very detailed and in order to use them as the ground truth, we manually segment them at a coarse level. However, it is possible to ask the user what level of details he/she prefers. Figure 4.1 shows the anatomy models for these animals. Figure 4.2 shows the corresponding ground-truth segmentations. Figure 4.3 presents the ground-truth of giraffe at a finer level of detail.

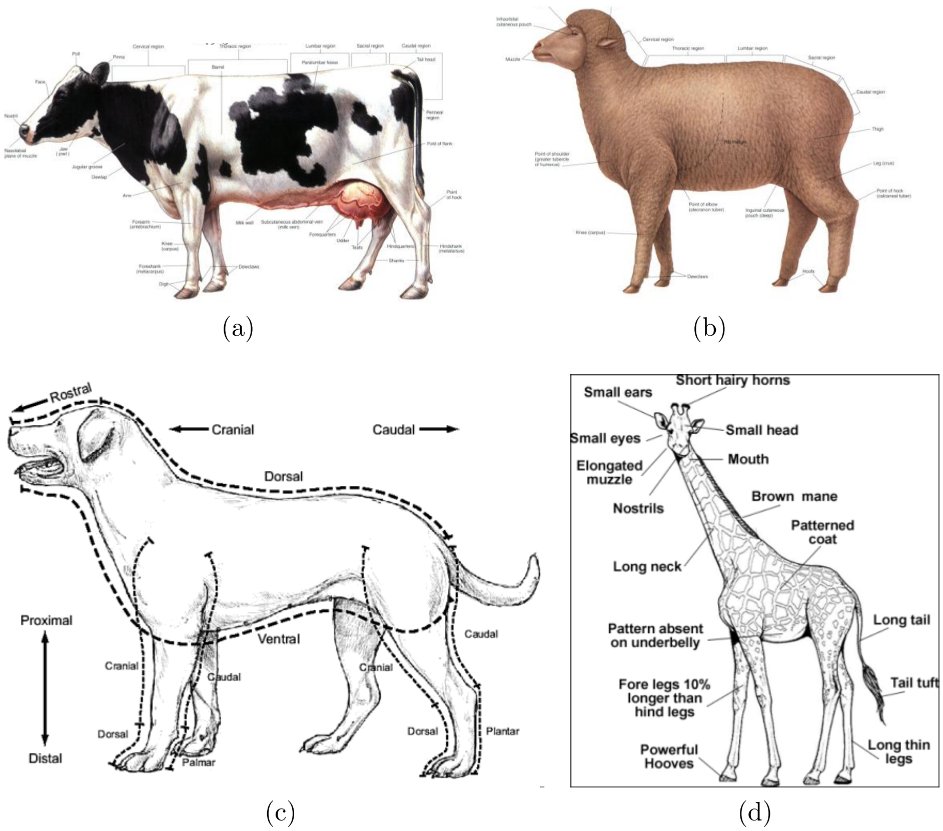


Figure 4.1: The anatomy model of (a) cow [75], (b) sheep [75] , (c) dog [74] and (d) giraffe .

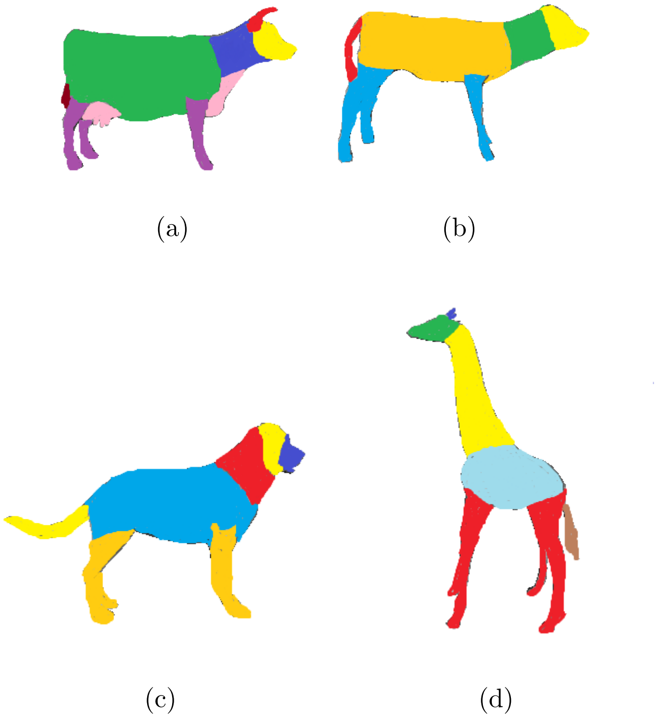


Figure 4.2: The manual segmentation for (a) cow, (b) sheep, (c) dog and (d) giraffe based on their anatomy.

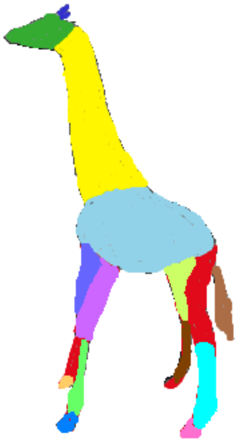


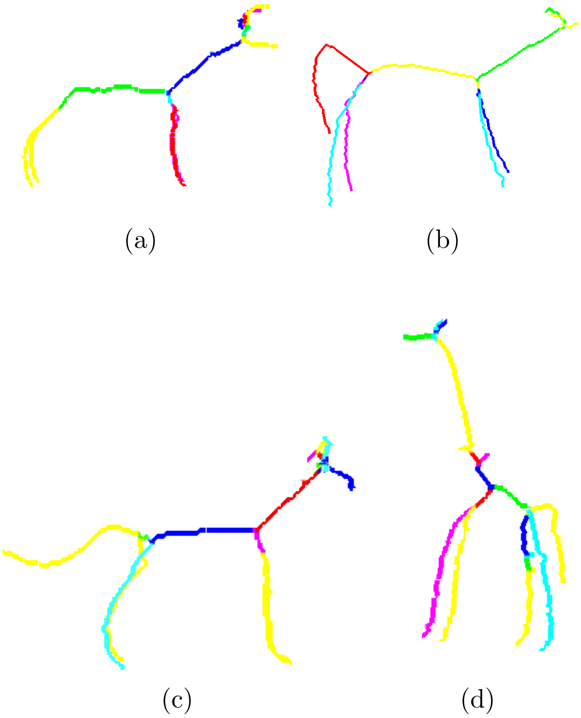
Figure 4.3: Detailed segmentation for the giraffe.

Table 4.1: Metric distance between different segmentation methods and groundtruth.

	RW	**RC**	**SDF**	**Skeleton**
Cow	0.77	0.71	0.65	0.57
Horse	1.00	0.84	0.79	0.45
Dog	0.8	0.75	0.25	0.65
Giraffe	0.64	0.43	0.57	Too noisy
Detailed giraffe	0.48	0.58	0.28	0.79

4.4 Results

Using the method proposed in [100], we extracted the skeletons and performed 3D segmentation for cow, sheep, dog and giraffe. Figure 4.4 shows the extracted skeletons. Figure 4.5 compares the segmentation results with random walk (RW) [62], random cuts (RC) [2], shape diameter (SDF) [99] and the ground truth. Note that the extracted skeletons may still contain some noise or unwanted branches which can lead to poor segmentation quality. One technique that can be used to eliminate the noise and improve the generated skeletons is scale space filtering. In future work we will use this method and other techniques to improve the results. Table 4.1 shows the metric distance between the segmented models and the corresponding animal anatomies. It is based on manual registration of the animal anatomy and generated segments and calculating the Euclidean distance for each segment, plus any extra segments that the methods generated. Our method as well as SDF [99] produce the best results. It is interesting to notice that with our anatomically defined ground truth, SDF [99] is better than random cuts [2]; this contradicts the evaluation based on the Princeton segmentation benchmark [22], which is entirely human defined. This confirms that human segmentations may not necessarily be a good means of defining ground truth for some applications.



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

Figure 4.4: Extracted skeletons for (a) cow, (b) sheep, (c) dog and (d) giraffe. Different colors are used for different segments of the skeleton.

4.5 Conclusion

We described an approach to obtaining ground truth for animal model segmentation based on expert knowledge, and used this information to compare various approaches. In future work, we will extend our study using a larger collection of animals. We will consider extracting the ground truth at different levels of detail. Furthermore, we will study how current algorithms can be extended to segment at varying levels of detail. One modification that can be done in the future is to let the user define the level of detail, or at least let the program know if the user needs a more detailed or a coarse segmentation. Then, a program can generate the skeleton and segmentation according to the user requirements.

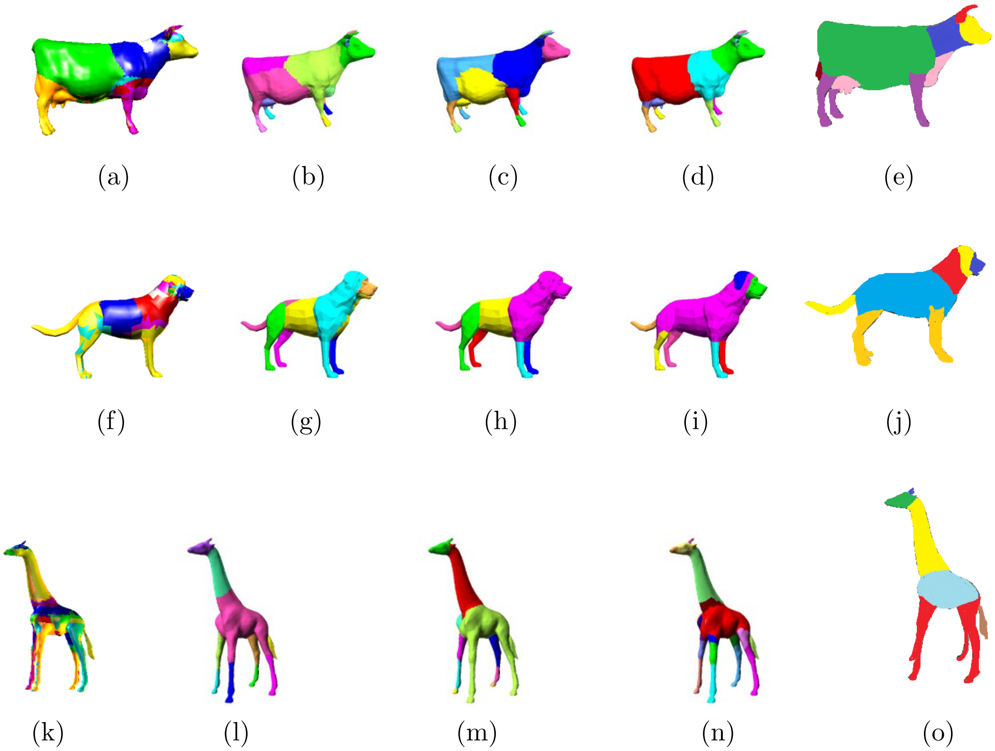


Figure 4.5: Comparison of the proposed method with random walk, random cut, shape diameter and ground truth. The first column is our skeleton-based method, the second column corresponds to random walk, the third column corresponds to random cuts, the forth column corresponds to shape diameter and the last column is the ground truth. Top row shows segmentation for the cow, middle for the dog, and bottom for the giraffe.

Chapter 5

Automatic Animation Skeleton Extraction and Model Transfer

Automatic 3D skeletonization is an important and challenging problem in computer graphics. It can be used in several applications; such as animation, mesh decomposition and 3D segmentation. To animate an articulated 3D character an animation skeleton needs to be extracted from or be embedded into a 3D model so that the model can be automatically deformed during animation. In conventional animation software, this process is mostly done manually by expert animators, which makes it a very tedious and time consuming step. Recently, some automatic rigging approaches have been proposed in the literature. However, most of these techniques are not fully automatic and require a front facing model with neutral T-pose to accurately extract or embed the animation skeleton. In this chapter we propose a fully automatic skeleton extraction approach based on optimization of constraints on human shape that can generate the animation skeleton, regardless of the model's orientation and position. Given an arbitrary 3D model and a template skeleton, the algorithm first extracts the 3D curve skeleton and then optimizes a penalty function to find the best match for the positions of the model's joints. Experimental results demonstrate the effectiveness of our approach.

5.1 Background and Related Work

Improvements in 3D capture technologies have led to many 3D models being widely accessible to the general public. Now, people would like to bring these articulated 3D models to life easily and quickly, by animating them automatically. There are two main techniques used in the literature to automatically animate a 3D character. The first one is skeleton-based animation [13] which is mostly used in conventional animation software, and the second one is mesh deformation [113]. The most important step for skeleton-based animation is rigging, which means embeding a skeleton inside the 3D character and associating the mesh vertices to the skeleton bones. Now-a-days conventional animation softwares require extensive manual effort to rig the model. This process is very tedious and time consuming, even for expert animators. Motion retargetting is the next step in animating the 3D model, which implies adapting the animated motion from one articulated figure to another. One of the first work in this area was by [37]. They assumed that two figures have the same topological structure but different physical dimensions. There are also several studies on motion synthesis. For this problem researchers try to create new motion based on similar motions in a motion database. Several constraints have to be met so that the final results look realistic [43]. The final step is skin attachment where the mesh vertices are attached to the skeleton bones based on their weights. Several linear skinning techniques have been proposed in the literature [46]. Jacobson et al. [47] proposed a non-linear method that performs the automatic skinning very fast and efficiently. Another challanging problem in the area of skinning is managing the clothes and animating them naturally. One of the first works in this area was proposed by [114]. For this chapter we just used the plain 3D models without any textures and clothing. In future work we can add automatic animation of the clothes as well.

Despite recent interest and studies in automatic rigging, this process needs to be done mostly manually to obtain satisfactory results. The techniques proposed in the literature require a front-facing, T-pose 3D model to extract or embed an animation skeleton. Because of this restrictions, existing techniques are not fully automatic. To address this limitation, we present a new method to automatically extract an animation skeleton for an arbitrary 3D model, regardless of its position and orientation. First, we extract a curve skeleton. Then, we label a set of skeleton points as key points or joint candidates. Finally, we minimize a penalty function to refine the set of key points. The penalty function uses template skeleton information and contraints on human shape based on anatomical and 3D mesh information to accurately find the final set of joints which form the animation skeleton.

The remainder of this chapter is organized as follows: Section 5.2 talks about different steps required in animation pipeline and the related research for each step. Sections 5.3 explains the proposed method and the optimization framework which is used to automatically extract an animation skeleton from an aribtrary human model. Section 5.5 discuss how to transfer a non T-pose model to a T pose model. Section 5.6 describes the experimental setup, shows some results and compares the proposed method with the state of the art techniques [13] proposed in the literature. Finally, the last section concludes the chapter.

5.2 Animation Pipeline

In this section we talk about the background of animation skeleton, motion targeting and motion generation and skin deformation.

5.2.1 Automatic Creation of the Animation Skeleton

Since the skeleton joints are the control mechanism in animation, we start by reviewing two skeleton-based animation approaches commonly used in the literature. The first is *skeleton embedding*, and the second is *skeleton extraction*. The former relies on repositioning and resizing a template animation skeleton to fit inside a 3D character. The latter aims at extracting a model specific curve skeleton from the 3D character, using a 3D skeletonization algorithm. The extracted curve skeleton is then converted into an animation skeleton, either by using anatomical information or matching a template animation skeleton. However, all current skeleton embedding and extraction techniques that have been proposed in the literature are not fully automatic and often require the 3D character to be front-facing and in a neutral, T-position. Otherwise, manual adjustment of skeleton joints is necessary.

Skeleton Embeding

An early work that automatically embedded a template animation skeleton into a 3D character was proposed by Baran et al. [13]. The authors formulated the joint positioning problem as an optimization problem and computed the joint positions inside the character by minimizing a penalty function. However, they assumed that the 3D model is appropriately proportioned and in the same position and orientation as the template skeleton. Therefore, if for instance the character is in a sitting position or upside down, their algorithm will not produce desirable results because the template skeleton used is posed differently.

Another skeleton embedding technique was proposed by Poirier et al. [88], where the extracted curve skeleton information was used to guide embedding the template animation skeleton inside the 3D model. The head node was first selected manually on the curve skeleton. The program then matched the segments of the curve skeleton, which were fitted into the template skeleton using the symmetry information and geodesic distance from the head to every other node. This approach is semi-automatic and as the authors pointed out if the pose deformation is not isometric, the algorithm cannot ensure the correct harmonic graph.

Skeleton Extraction

3D skeletonization has been extensively studied in the literature. Cornea et al. [26] presented a survey covering different skeleton extraction techniques. Developers usually use clean and thin curve skeletons for animation. In this section, some previous papers on automatic animation based on skeleton extraction are reviewed.

One paper in this area was proposed by Teichmann et al. [107]. The authors extracted a skeleton hierarchy from a 3D polygonal mesh for animation. They first extracted the medial axis of the object using a Voronoi skeleton, then some of the skeleton key-points were selected by the user. The selected points correspond to the end points of the animation skeleton. This method is not fully automatic and relies on precise selection of some joints. In another method a voxelized skeleton was extracted, and the skeleton branches were then approximated with straight segments [115]. The shared points between two segments are labelled as key-points (skeletal joints). Although this approach is quite fast and simple, the resulting skeleton may miss critical joints and cannot be matched accurately with the animation skeleton. For instance, in a human model with straight arms, elbow joints will be missed by their algorithm.

The repulsive force field approach was used by Liu et al. [67] to extract the animation skeleton. The algorithm shoots rays from points inside the model to calculate the repulsive force field magnitudes. The candidate points are labelled by finding the local minimal magnitude of the points. The extracted skeleton is then refined and undergoes thinning to generate the animation skeleton. Their method does not guarantee that the extracted skeleton will be compatible with the animation skeleton.

Some researchers used anatomical information to extract the animation skeleton, e.g., [11] and [84]. MoCap data often define joints, e.g., three joints along the spine, which may not have an anatomical meaning, but they are used for animation. In this case, the extracted skeleton may miss the joints that do not have anatomical correspondence. Also, in the method proposed by Aujay et al. [11] the user needs to select the root point manually, and in the method proposed by Pan et al. [84] the initial model needs to be in the resting position. Schaefer and Yuksel [96] extracted the animation skeleton from a set of example poses from the same 3D model. This is not a feasible approach if only one arbitrary 3D model is given for animation. Pantuwong and Sugimoto [85] presented an automatic rigging technique based on skeleton extraction and template matching. They first extracted the curve skeleton from the 3D model using a skeletonization algorithm. Then, they performed further processing to clean the skeleton and make sure the symmetric segments, such as both arms and both legs, are connected to the central axis or spine at the same location, and have approximately the same length and topology. In their junction point analysis step, they classified the extracted skeleton into different categories, e.g. bipeds, snakes and birds, in a database of animation skeletons. After retrieving the suitable template skeleton, they located the skeleton joints on the curve skeleton using anatomical information, but with the assumption that the topology and pose of the 3D model should be the same as the template skeleton.

5.2.2 Motion Retargeting and Motion Generation

The quality of the animation depends on the realistic motion, and generating such motion is quite expensive and challanging. People usually record the natural human movement of a live character through a motion capture system such as Vicon motion capture [112]. The captured motion will be transferred to new articulated characters (Motion Retargeting), or several captured motion can be combined together to generate new movements (Motion Generation).

Motion Retargeting

One of the first work in the area of motion retargetting was proposed by [37]. In this study they used the motion of an articulated model for another model with the same topology but different bone lengths. The same topology implies that the two models have the same number of the joints, the same connectivity of the bones and the same degree of freedom. Even with the same topology, two models cannot directly share the motion and some adaption or modification is required. Some features of the motion are quite important and should be maintained during the animation. For instance during a walking motion the feet should touch the floor regardless of model's size. In this study they identified the important properties of the motion and set them as constraints that should be maintained all the time. They later modify the motion during retargetting to make sure that the constraints are always valid. They used a space time optimization solver and consider the entire motion simultaneously, not the individual frames as used in inverse kinematics. The problem with adjusting the motion in each frame, to meet the constraints, is introducing the high frequency components to the original motion. The existance of high frequency components or lack of them in the motion should always be preserved in order to retarget the motion successfully. This method has no information about the motion and it highly relies on the constraints and if the solver is not provided with a comprehensive set of constraints the resulting animation may be unrealistic. On the other hand, [95] proposed a motion aware retargetting approach. They analyze and classify the motion to determine the motion strtucture and identify its constraints. This eliminate the step of manually defining the constraints. The methods discussed thus far retarget the motion, using an spacetime optimization approach; ie, consider the whole motion for retargetting. [106] proposed a motion editing method based on per-frame Kalman filter approach.

[68], [79] proposed a method to solve the motion retargetting problem for geometrically and hierarchically different models by using an intermediate skeleton. The intermediate skeleton has the same number of joints as the target skeleton, but it has the same orientation and position as the source skeleton. They used inverse kinematics to adjust the motion and manually set the corespondece between the two hierarchies. Another approach that retargets the motion between different topological models has been proposed by [12]. However they can only retarget the motion if the two models have different hierarchical structures and segments' length, and not between any two arbitrary models. [42] proposed a new system which records the motion in a generalized form and later the user can specialize it into different characters. This approach enables the user to retarget a morphology independent motion to models with completely different topologies.

One can also retarget the motion in the mesh level as proposed by [105]. This approach can transfer the motion between different meshes with different number of triangles and different connectivity. Here the user manually builds the correspondence map between the source and the target by selecting a small set of vertex markers. This paper does not retarget the motion in the skeleton level but rather directly deform the target mesh. Another research used physical properties such as segment mass, center of mass, velocity and so on to minimize the differences between source and target motion [110].

Motion Generation

The editting and retargetting of mocap data can be useful to adapt the available motion to different characters and models. However, motion editting won't be helpful if the required motion is different than the one already captured. In this case one should capture more motion which is a very expensive and time consuming step, which indicates the need for motion generation and motion synthesis. [59] used motion graph, which is a structure that enables the captured data to be reassembled in different ways. This graph contains both pieces of original motion captured and automatically generated motion. New motions can then be generated by building walks on the graph. Another approach that is used to generate new motion is motion cut and paste as discussed in [5]. Motion database is quite important to build a rich set of generated motions and behaviour [63]. In order to generate new movements, one can use a collection of similar movements. [60] used Principal Component Analysis (PCA) to extract the set of basis elements from existing human motion data and then used Hidden Markov Models (HMM) to find the optimal linear combination of basis elements to describe a natural generated movement.

5.2.3 Skin Deformation

Skin deformation based on an underlying skeleton is a common method to animate believable organic models. The most widely used skeletal animation algorithm, linear blend skinning, is also known as skeleton subspace deformation, vertex blending, or enveloping. It runs in real-time even on a low-end hardware but it is also notorious for its failures, such as collapsing and candy wrapper joints. To remedy these problems, one needs to formulate the non-linear relationship between the skeleton and the skin shape of a character properly, which however proves mathematically very challenging. Placing additional joints where the skin bends increases the sampling rate and is an ad-hoc way of approximating this non-linear relationship.

Kavan and Zara [55] presented an algorithm which removes these shortcomings while maintaining almost the same time and memory complexity as the linear blend skinning. Unlike other approaches, this method works with exactly the same input data as the popular linear version. This minimizes the cost of upgrade from linear to spherical blend skinning in many existing applications: the data structures and models need no change at all.

Yang et. al. [120] proposed a method that is able to accommodate the inherent non-linear relationships between the movement of the skeleton and the skin shape. They used the curve skeletons along with the joint-based skeletons to animate the skin shape. Since the deformation follows the tangent of the curve skeleton and also due to higher sampling rates received from the curve points, collapsing skin and other undesirable skin deformation problems are avoided. The curve skeleton retains the advantages of the current skeleton driven skinning. It is easy to use and allows full control over the animation process.

Different alternatives have been proposed in literature to overcome the artifacts of LBS. All of them successfully combat some of the artifacts, but none challenge the simplicity and efficiency of linear blend skinning. As a result, linear blend skinning is still the number one choice for the majority of developers. Kavan et. al. [54] presented a novel GPU-friendly skinning algorithm based on dual quaternions. They showed that this approach solves the artefacts of linear blend skinning at minimal additional cost. Upgrading an existing animation system (e.g., in a video game) from linear to dual quaternion skinning is very easy and has negligible impact on run-time performance.

5.3 Robust Human Animation Skeleton Extraction Using Compatibility and Correctness Constraints

The algorithms discussed thus far require the 3D models to be in the front facing T position, or they are not fully automatic. However, our method can work for an arbitrary 3D model with any initial position and orientation, and can automatically generate the animation skeleton. In this section a simple view of the proposed method is presented. Fig. 5.1 shows the steps in hierarchy. The novelty of our work lies in the optimization framework. We define a penalty function which considers two different cost function $\varphi_{Compatibility}$ and $\varphi_{Correctness}$.

$\varphi_{Compatibility}$ measures which joints would be the closest match to the template skeleton, based on the anatomical information of a human model. $\varphi_{Correctness}$ checks that the bones formed by the joints are inside the 3D mesh and does not intersect the mesh. Eq.(5.1) demonstrates the penalty function.

$$\varphi(n) = (1 - \lambda)\ \varphi_{Compatibility}(n) + (\lambda)\ \varphi_{Correctness}(n) \tag{5.1}$$

where n is a skeleton point and $0 \leq \lambda \leq 1$ is the weight for the cost components. The details of the two terms $\varphi_{Compatibility}$ and $\varphi_{Correctness}$ and λ will be explained in Section 5.3.4. Fig. 5.2(a) shows the situation where the selected key point in red circle is in the wrong position inside the mesh. The bones connected to that keypoint intersect the mesh which is an indication of the wrong position. Fig. 5.2(b) shows the correct position of the same point and there is no intersection between the bones and the mesh.

There are four step in the overall process: (i) Computing a Template Skeleton from the Motion Capture (MoCap) file; (ii) Extracting a Curve Skeleton of a given 3D model that needs to be animated; (iii) Generating an Animation

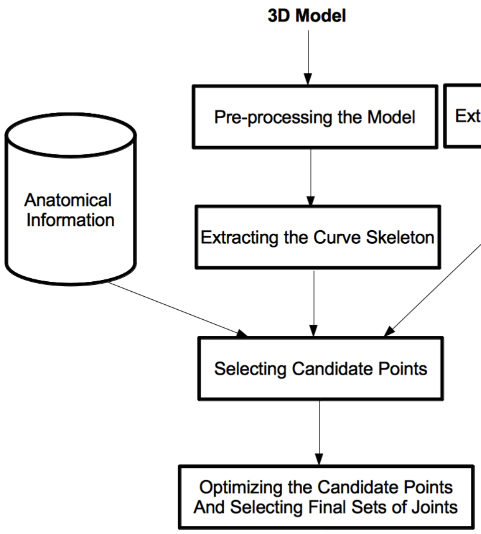
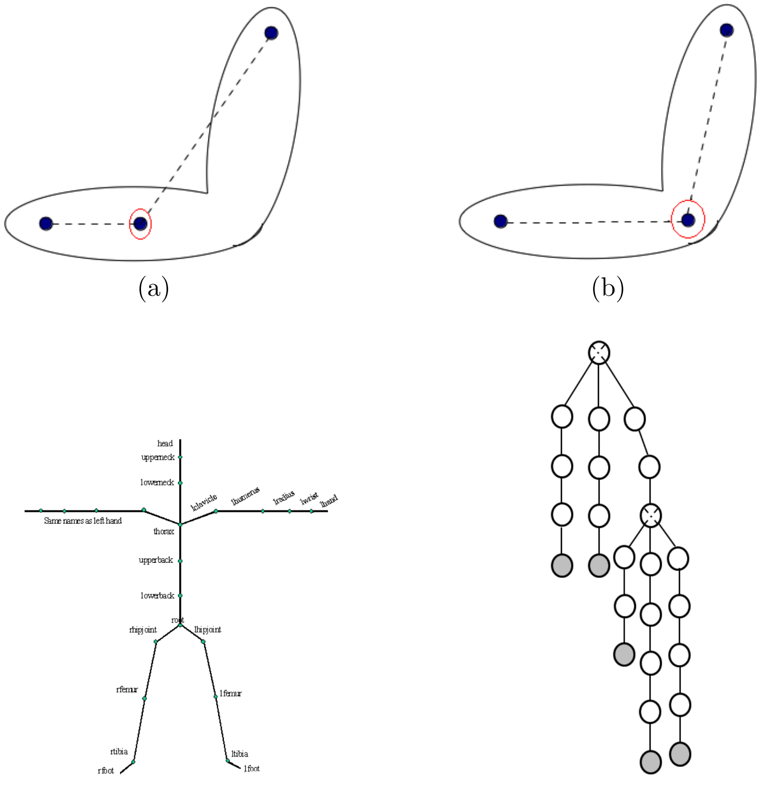

3D Model
MoCap File
Pre-processing the Model
Extracting the Template Skeleton
Anatomical Information
Extracting the Curve Skeleton
Selecting Candidate Points
Optimizing the Candidate Points And Selecting Final Sets of Joints

Figure 5.1: The general view of the proposed method.

Skeleton, which is the Curve Skeleton Adapted to make it compatible with the Template Skeleton, and suitable for animation; and (iv) Computing the optimized animation skeleton so that it reduces artefacts during animation.

5.3.1 Extracting the Template Skeleton

The first step is getting the template skeleton. Usually MoCap files contain two parts. The first part is the skeleton hierarchy which gives information about the number of joints, the location of the joints relative to their parents, the length of the bones, the position and orientation of the bones and so on. For a biped model there are usually 28 joints in a MoCap file. Some of these joints do not have any anatomical correspondence. The second part in a MoCap file contains the motion information of joints in each frame. Usually the motion is represented through rotations in the x, y and z axes. For a human like model the root is the pelvis joint. Fig. 5.2 (c) shows a sample of a template skeleton



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

Figure 5.2: ((a) The key point is wrongly positioned inside the mesh and the bones connected to that point intersect the mesh; (b) The key point is in the right position and there is no intersection between the bones and the mesh; (c) A human template animation skeleton extracted from a MoCap file; and (d) The tree structure.

extracted from a MoCap file. The tree structure of the template skeleton is shown in Fig. 5.2 (d). Note that there are three types of joints in each skeleton and correspondingly in each tree. The end points (e_T) of the skeleton, with the degree of one ($deg(e_T) = 1$), form the leaves of the tree which are shown in gray. For instance in Fig. 5.2(c) the nodes *head,rhand, lhand, rfoot* and *lfoot* are the end points. The nodes that have three or more connections are shown with cross and correspond to the junction points (j_T) of the skeleton. The junction points have $deg(j_T) \geq 3$. The nodes *root* and *throax* in Fig. 5.2 are the junction points. The rest of the nodes are regular skeleton joint points (n_T), with $deg(n_T) = 2$. Each of these points has a value $d(n_T)$ which is the normalized bone length. Eq. (5.2) describes the normalized bone length, where $d(n_T, e_T)$ is the Euclidean distance between the joint n_T and its adjacent end point e^T in the hierarchy and $d(j^T, e^T)$ is the Euclidean distance between the same end point and its adjacent junction point j_T in the hierarchy. In other words the normalized bone length is the proportional length of a bone segment to the entire limb. For instance the proportional length of the forearm to arm.

$$d(n_T) = \frac{d(n_T, e_T)}{d(j_T, e_T)} \tag{5.2}$$

5.3.2 Extracting the Curve Skeleton

In the second step a curve skeleton is extracted from the 3D model. In theory any 3D skeletonization algorithm can be used to get the curve skeleton. 3D skeletonization techniques have been widely studied in the literature. A survey of different skeletonization approaches is given in [26]. Curve skeleton extraction techniques can be broadly classified into two groups [9]: geometric (if the algorithm is based on a surface) or volumetric. However, for generating an animation skeleton, we would like a unit-width curve skeleton where all

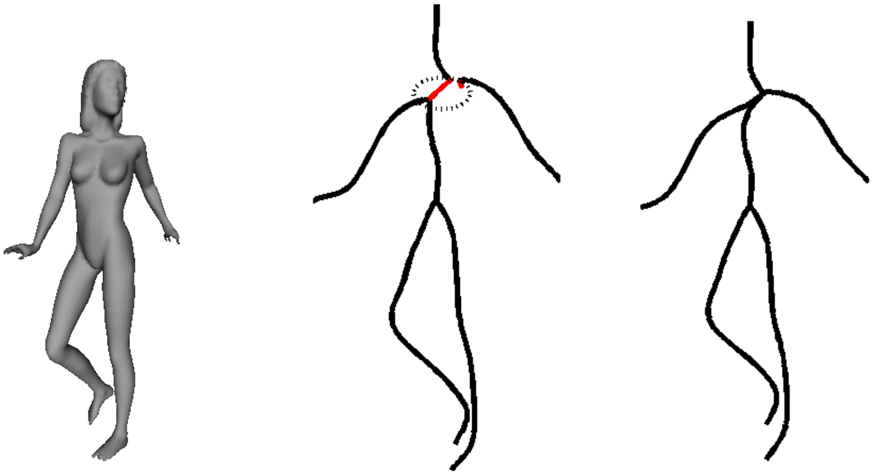


Figure 5.3: (Left) A 3D triangular mesh. (Middle) Extracted unit-width skeleton using VNSM technique. (Right) Symmetric, smooth and clean curve skeleton.

the components are connected. Therefore, we adapt the Valance Normalized Spatial Median (VNSM) approach proposed by [117]. Although the extracted curve skeleton is a connected unit-width skeleton, it is very sensitive to noise and may contain unwanted branches. Fig. 5.3(a,b) shows an example of a 3D triangular mesh from [23] and the extracted skeleton using the approach proposed in [117]. Undesirable branches are shown inside the circle. Also, the left and right arms may be connected to the spine in several different positions which contradicts the symmetry property of human models. To remove the unwanted branches we use the branch length information ($l(b_{Si})$), and remove any branch shorter than a threshold. Eq. (5.3) defines the constraint for removing the unwanted branches, where q is the number of branches. We do not consider finger and toe branches, since the MoCap file does not have details at this level.

$$threshold = \frac{\sum_{i=1}^{q} l(b_{Si})}{3 \times q} \tag{5.3}$$

Table 5.1: The threshold values for joint positioning.

joint	th_1	th_2	th_3
hipjoint	0.1	0.15	0.13
femur	0.47	0.53	0.5
tibia	0.87	0.93	0.9
clavicle	0.26	0.32	0.29
elbow	0.59	0.65	0.62
wrist	0.84	0.90	0.87

5.3.3 Automatic Selection of candidate Points

In order to speedup the process of optimization and finding the final set of joints, we first automatically select a set of candidate joints or key points (kp) from the curve skeleton based on three different factors, and later optimize these points:

$$kp = \{n_S | j_S \ or \ e_S \ or \ th_1 \leq d(n_S) \leq th_2\} \tag{5.4}$$

where n_S is the curve skeleton point, $d(n_S)$ is the normalized bone length, and th_1 and th_2 are the upper and lower limits for the position of a regular joint. The skeleton end points (e_S), the skeleton junction points (j_S) and the points within the threshold limits are selected as key points. Fig. 5.4 shows an example of the selected candidate points. The values for th_1 and th_2 are shown in Table 5.1. These values have been obtained by analysing 144 different subjects presented in the CMU MoCap databse. The minimum normalized bone length for each joint forms th_1, the maximum normalized bone length forms th_2 and the average forms th_3. Note that from anatomical point of view there is always a relationship between the proportional bone length for limbs as described in [87]. For instance the length ratio of femur to tibia and humerus to ulna has been reported to be 1.21 and 1.22 respectively with a standard deviation of 7%. This study was conducted among 24 men and 14 women with different origins and age groups.

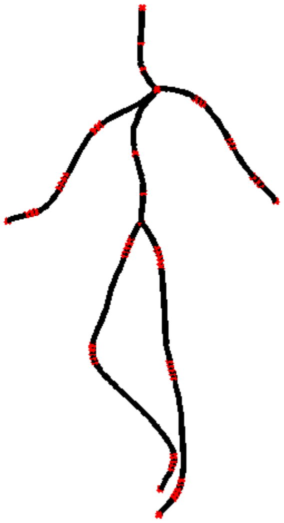


Figure 5.4: The set of selected candidate points.

To compare the curve skeleton with the template skeleton, we use a tree structure for both curve and template skeleton. The template skeleton is already presented in the tree form. To generate the tree for the curve skeleton, we first need to find the root. For a human-like model, the pelvis is selected as the root. The *pelvis* is the curve skeleton junction point (j_S) with degree equal to three, while the other junction point, *throax*, has a degree of four. Curve skeleton end points (e_S) are the leaves of the tree and the rest of the key points are tree nodes. The tree is generated based on the connectivity between points. We compare the two trees branch by branch and based on their leaves. Therefore, labelling the end points of the curve skeleton, which forms the leaves of the tree, is a crucial step. To label these end points we use the following equation. This equation works for human-like skeletons without loops:

$$label(e_S) = \begin{cases} feet \text{ if } deg(j_S) = 3 \\ head \text{ if } deg(j_S) = 4 \\ \quad \text{and } length = min(branch.length) \\ hand \text{ otherwise} \end{cases} \tag{5.5}$$

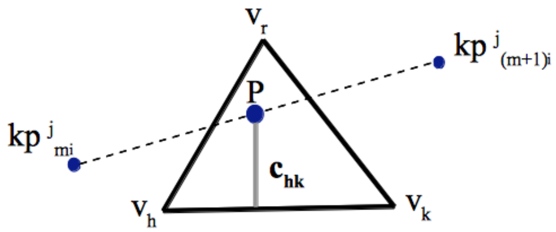


Figure 5.5: Geometric illustration of the variables.

5.3.4 Optimization Framework

Once we have the hierarchy for the candidate points, the last step is to match them with the template skeleton and refine them based on the mesh information to get an accurate set of final joints. We would like to find the best match, therefore an optimization approach is used to minimize a penalty function as shown in Eq.(5.1). The penalty function (φ) is based on two important cost functions. First is $\varphi_{Compatibility}$, which means the final set of joints should have accurate proportions and be compatible to the template skeleton. For better understanding of the algorithm and without losing generality, suppose we have M different tree nodes or joint positions in both template skeleton and curve skeleton and M' different candidate points, kp, for each joint position in the curve skeleton as shown in Fig. 5.4. Analysing each branch separately, we calculate the penalty values for each candidate points i in the joint position m, which is (kp_{m_i}). The *Compatibility* cost function is explained in Eq.(5.6).

$$\varphi_{Compatibility}(kp_{m_i}) = |d(n_{T_m}) - d(kp_{m_i})| \tag{5.6}$$

Note that we would like to find the best match of a candidate point, kp_{m_i}, to a specific template skeleton joint, n_{T_m}. Therefore, we fix joint m of the template skeleton and search for the best candidate point in the same joint position m in the curve skeleton kp_{m_i}. The second factor is $\varphi_{Correctness}$, which means the two bones that are connected to the selected candidate points should not intersect the 3D mesh. Otherwise, the selected candidate point is not in the correct position, as can be seen in Fig 5.2 (a). The *Correctness* cost function is demonstrated in Eq.(5.7).

$$\varphi_{Correctness}(kp_{m_i}) = f(kp_{m_i}, kp_{(m-1)_{i'}}) + f(kp_i, kp_{(m+1)_{i''}}) \tag{5.7}$$

where function $f(a, b)$ checks if the bone ab intersects the mesh, what is the distance between the intersected point, P, and the edges of the faces it intersects . Each candidate point kp_{m_i} can be connected to the candidate points in two consecutive joint positions, $kp_{(m-1)_{i'}}$ and $kp_{(m+1)_{i''}}$. We have to check if any of the two bones intersect the mesh or not. Function f is defined in Eq.(5.8)

$$f(kp_{m_i}, kp_{(m-1)_{i'}}) = \sum\nolimits_P \frac{|\{v_h, v_k\} \times \mathbf{c_{hk}}|}{|\{|v_h, v_k\}|} + \frac{|\{v_h, v_r\} \times \mathbf{c_{hr}}|}{|\{|v_h, v_r\}|} + \frac{|\{v_k, v_r\} \times \mathbf{c_{kr}}|}{|\{|v_k, v_r\}|} \tag{5.8}$$

In the 3D mesh v_h, v_k and v_r form a triangle face, and $\{v_h, v_k\}$ is the edge between triangle points v_h and v_k. If the bone that is formed by two candidate points in two consecutive joint positions kp_{m_i} and $kp_{m+1_{i'}}$, intersects the mesh, then we call this intersection point P. This is a Ray-triangle intersection problem which has been described in [78]. We just need to check that P is within the bone segment. The vector between P and $\{v_h, v_k\}$ is denoted by $\mathbf{c_{hk}}$. Fig. 5.5 describes the variables visually.

Note that each candidate point i has M'^2 penalty values considering the different combination of the consecutive joints. Starting from the leaf of each branch (e_S) in the curve skeleton, we iteratively select a candidate point, in joint position m, with minimum penalty value and then choose the consecutive joints accordingly. Lets say the minimum penalty value for joint position m is for the candidate point i, kp_{m_i}. The selection of i leads us to the candidate points i' and i'' in the joint positions $m-1$ and $m+1$ respectively. Since they are the other joints of the bone that created the minimum penalty function. This goes forward until we traverse one branch completely. Finally, we calculate the penalty path for all the different combinations and choose the path with minumum penalty as the optimal path. The candidate points that form the optimal path are the final optimized joints. Eq. (5.9) explains the optimization process and how to select an optimum key point *op*.

$$op_m = \min(\min_i(\varphi(kp_{m_i})) + \sum_{o=1}^{m-1} \varphi(kp_{oi'|i}) + \sum_{u=m+1}^{M} \varphi(kp_{ui''|i})) \tag{5.9}$$

Note that in Eq. (5.9) we first choose the minimum penalty value for the candidate point kp_{m_i}, and then we choose the path with the minimum total penalty to find the final set of optimized joints.

In Eq.(5.1) each cost function has a weight λ and $1-\lambda$, where the weight has a value between $[0,1]$. We performed some user studies to find the best value for λ. We changed the values of λ in step of 0.1 and showed the users the generated animation skeleton superimposed inside the mesh. We then asked the user to select a model with the skeleton best fitted inside the mesh and the joints are correctly positioned . The result of the user study indicates that the best value for λ is 0.8, which means that $\varphi_{Correctness}$ has a weight of 0.8 and $\varphi_{Cmpatibility}$ has a weight of 0.2. This implies that the most important factor is $\varphi_{Correctness}$. In other words a key point is wrongly positioned if it intersects the mesh, so we would like to avoid it happening by giving the term *Correctness* a higher weight. However, there still can be human models that do not follow the anatomical rule precisely, therefore we give the *Compatibility* term a lower weight. Fig. 5.6 shows the final result of the selected joints and the generated animation skeleton.

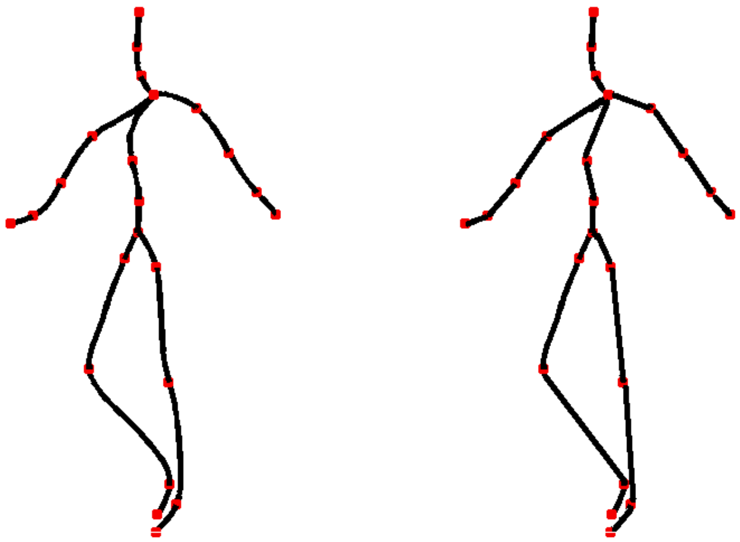


Figure 5.6: (Left) The set of final joint points. (Right) The final animation skeleton.

5.4 Automatic Anatomical Aware 3d Skeletonization

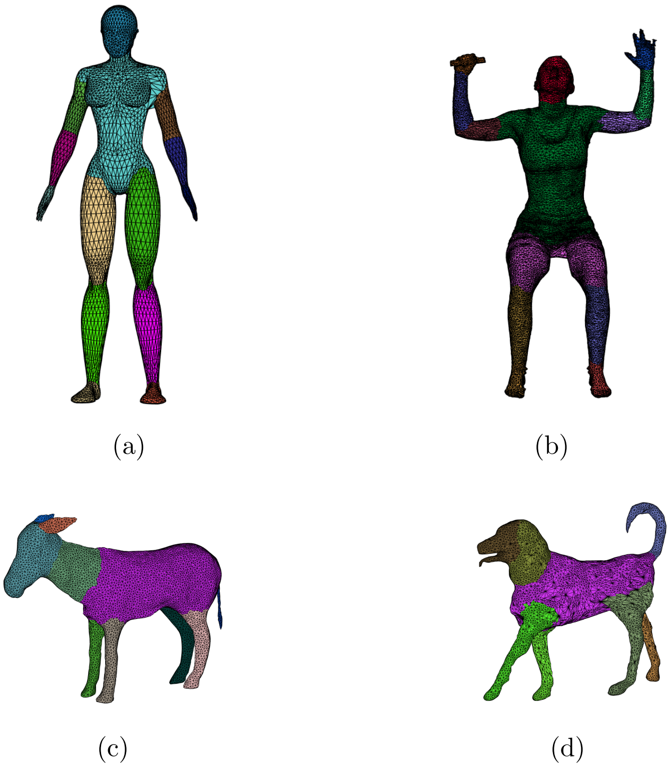
Mesh segmentation and 3d skeletonization of a biological model, such as bipeds and quadrupeds, can be considered as related problems, specially if both applications use the anatomical features of the underlying model.

Previous studies [38], [50], [69] showed that one can use a segmented mesh to extract curve skeleton or segment the mesh based on extracted skeleton respectively. Even after creating the skeleton based on mesh segmentation, the location of the joints can still be vague.

In this thesis we first segment the mesh based on anatomical information as proposed by [50], then find the borders of the segmented regions and finally locate the joints.

5.4.1 Mesh Segmentation

Mesh segmentation is decomposing a polygonal mesh or 3d volume into different segments based on geometrical or semantical features. Geometrical



(a)

(b)

(c)

(d)

Figure 5.7: Mesh segmentation for two human models and two animal models, different colors shows different labels.

features include curvature, surface normal and distance to a plane, while semantical features can be anatomical properties of biological models for instance. [98] presents a comprehensive study on mesh segmentation and different geometrical and semantical techniques. As [49], [50] discussed, a reliable way to segment a mesh semantically is through learning segments based on some training dataset.

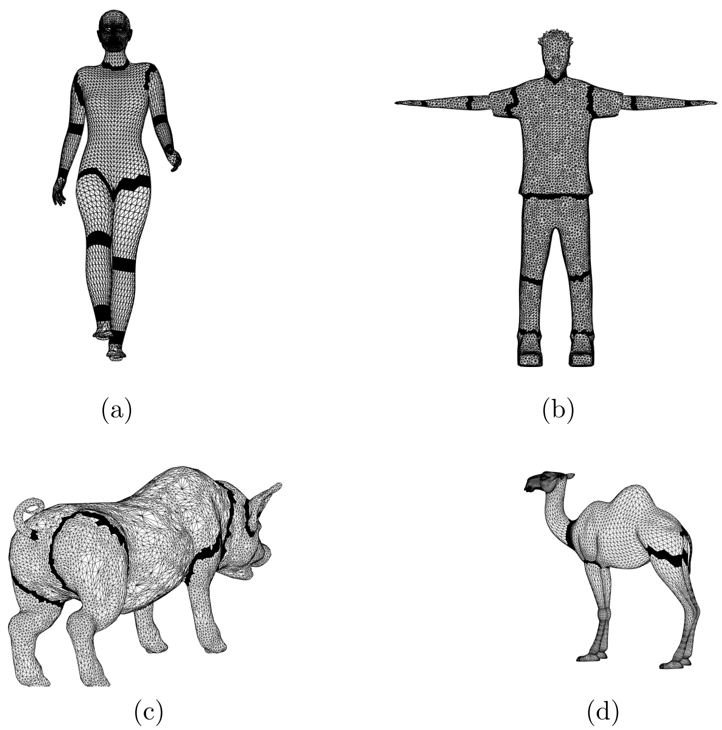
In this thesis we used the Conditional Random Field (CRF) model for mesh segmentation as described by [50]. Each face f_i of the mesh has a unique feature vector $\boldsymbol{v_i}$ which includes geometrical properties such as face curvature, shape diameter and average distance from medial axis. Also adjacent faces i and j have a binary feature vector $\boldsymbol{u_{i,j}}$. This includes derivative of face curvature, difference between shape diameter and difference between distance from medial axis. The binary feature vector implies if two adjacent faces should belong to the same segment or they belong to different segments. In order to label each face, the model should minimize Equation 5.10.

$$E(l,\theta) = \sum_i a_i E_1(l_i; \boldsymbol{v_i}, \theta_1) + \sum_{i,j} |e_{i,j}| E_2(l_i, l_j; \boldsymbol{u_{i,j}}, \theta_2) \tag{5.10}$$

where a_i is the area of each face i and $|e_{i,j}|$ is the length of edge between two adjacent faces i and j. θ is the parameter of the model and l_i is the label of face i from a set of predefined labels L. E_1 is the corresponding energy function for unique feature as explained in Equation 5.11. E_2 is the corresponding energy function for binary feature which penalize neighbouring faces that being assigned different labels.

$$E_1(l; \boldsymbol{v}, \theta_1) = -logP(l|\boldsymbol{v}, \theta_1) \tag{5.11}$$

$$E(l,\theta) = \sum_i a_i E_1(l_i; \boldsymbol{v_i}, \theta_1) + \sum_{i,j} |e_{i,j}| E_2(l_i, l_j; \boldsymbol{u_{i,j}}, \theta_2) \tag{5.12}$$



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

Figure 5.8: Detected borders for two biped and two quadruped models.

To find the best parameter we took the same approach as [50]. Figure 5.7 shows the result of mesh segmentation on a biped and a quadruped model. Note that based on training data one can achieve different level of details for mesh segmentation.

5.4.2 Border Detection

After segmenting the meshes based on anatomical cue, we should detect the border faces between two adjacent segments. Note that during mesh segmentation each face has been labelled. Thus, to find the border faces we should check what segments its vertices belong to. In other word face f is a border face if at least one of its vertices is shared between two segments. Equation 5.13 summarizes this condition.

$$\{f \in B|(f^k \in s_i \ \ and \ \ f^k \in s_j)\} \tag{5.13}$$

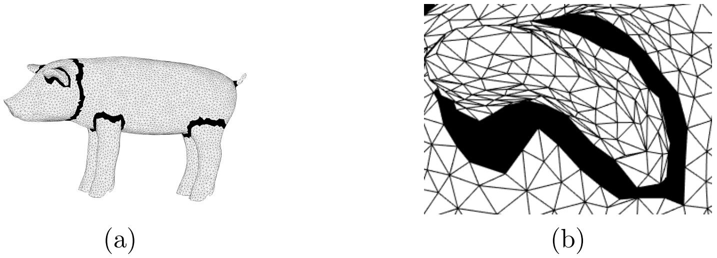

(a)
(b)

Figure 5.9: (a) concave borders of a pig model. (b) close up of the concave ear border.

where B is the set of border faces, s_i and s_j are two neighbour segments and f^k is the k'th vertex of face f.

Figure 5.8 shows some examples of the detected borders. Note that animals are segmented in a higher level of details and therefore the borders for upper limbs and lower limbs are not detected.

5.4.3 Joint Location

Joints are usually located on the borders of the segmented mesh, since each segments belong to different parts of human or animal body based on anatomical features. Since we can have a different level of details during mesh segmentation not all anatomical joints might be detected for a coarse segmented mesh for example. However we don't always need all this fine level of details skeleton. The mesh itself is not always high resolution and we usually don't have the details of hand of a human model. Besides finding all the anatomical joints of palm is not always necessary for animation or entertainment purpose for example.

The exact location of the joints inside the border region can be more challenging. The criteria is that the joints should be inside the model and as close as possible to the central line. One possible solution can be the centroid location. It is obvious that centroid of any convex shape is inside the shape and the best approximation for the centre of mass (COM) of the shape as other researchers used [14], [50]. Although the triangular faces are always convex, there is no guarantee that the border region is a convex shape as Figure 5.9 shows. Equation 5.14 shows how to calculate the COM based on the centroid of faces belong to the border.

$$c = \frac{\sum\limits_{f \in B} p_f}{n} \tag{5.14}$$

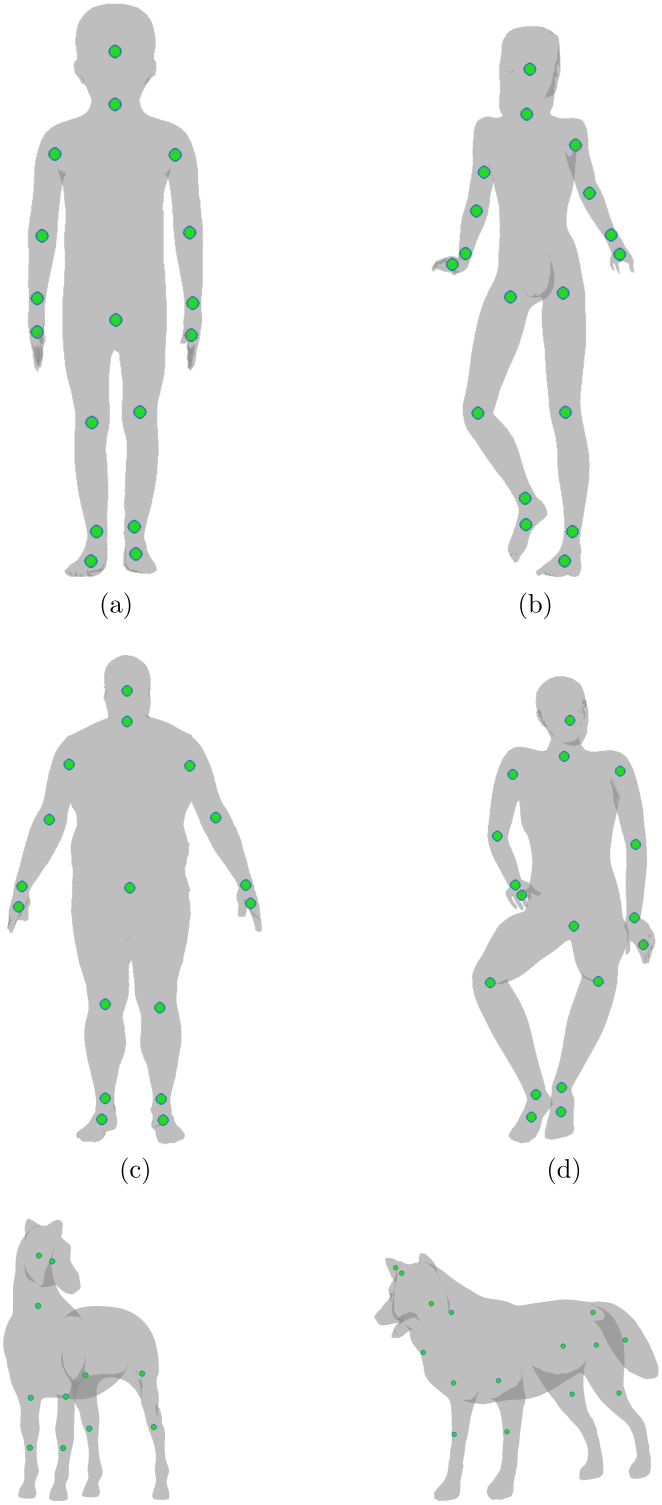
where n is the number of faces belonging to the border and p_f is the centroid of each triangular face.

Figure 5.10 shows the position of final joints for different biped and quadruped models. Note that level of details is different for human models comparing to animal models.

5.5 Transferring and Animating a non T-pose Model to a T-pose Model

Non T-pose animation is a technique that aims to generate natural transformations between any non T-pose skeletons to the neutral T-pose skeleton. It is not always easy to extract or embed a T-pose animation skeleton into a 3D human model due to its initial position. This is more problematic for the human models obtained by 3D scanning, especially models of babies and kids. In addition, transforming a non T-pose to a T-pose requires a large amount of calculations. Hence, many commercially available software do not provide efficient methods to standardize non T-pose skeletons. Here, we focus on developing a simplified transformation method, which enables skeletons in arbitrary poses to be standardized and used in other media conveniently.

Improvements in 3D capture technologies have led to many 3D models being widely accessible to the general public. Now, people would like to bring these articulated 3D models to life easily and quickly, by animating them



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

Figure 5.10: Final joints for biped and quadruped models.

automatically. However, the quality of the animation created depends largely on the the realistic motion, and generating such motion is quite expensive and challenging. To get the best results, people usually record natural human movement of a live character through a motion capture system such as Vicon motion capture [112]. The captured motion is transferred to new articulated characters (Motion Retargetting), or several captured motion are combined together to generate new movements (Motion Generation). To get a realistic animation the source and target models should have the same initial position which is not always possible. This problem is more severe if the target model is for a baby or kid obtained through 3D scanning as they cannot always maintain the neutral T-pose. One of the first work in the area of motion retargetting was proposed by [37]. In this study they used the motion of an articulated model for another model with the same topology but different bone lengths. Even with the same topology, two models cannot directly share the motion and some adaption or modification is required. Some features of the motion are quite important and should be maintained during the animation. They set some constraints manually and modified the motion during retargetting to make sure that the constraints are always valid. This method has no information about the motion and relies on the constraints. Thus, if the solver is not provided with a comprehensive set of constraints the resulting animation may be unrealistic. On the other hand, [95] proposed a motion aware retargetting approach. They analyze and classify the motion to determine the motion structure and identify its constraints. This eliminates the step of manually defining the constraints. The editing and retargetting of MoCap data can be useful to adapt the available motion to different characters and models. However, motion editing does not help if the required motion is different than the one already captured. In this case one should capture more motion which is a very expensive and time consuming step. This motivates the need for motion generation and motion synthesis. [59] used motion graph, which is a structure that enables the captured data to be reassembled in different ways. This graph contains both pieces of original motion captured and automatically generated motions. New motions can then be generated by building random walks on the graph. Another approach that is used to generate new motion is motion cut and paste as discussed in [5]. Motion database is quite important to build a rich set of generated motions and behaviour [63]. In order to generate new movements, one can use a collection of similar movements. [60] used Principal Component Analysis (PCA) to extract the set of basis elements from existing human motion data and then used Hidden Markov Models (HMM) to find the optimal linear combination of basis elements to describe a natural generated movement.

5.5.1 Proposed Method and Details

To animate a 3D human model one needs to have an animation or an IK skeleton, as shown in Fig. 5.2(c), which can be in different formats. Regardless of its format, the IK skeleton always contains information about the number of the joints, the location of the joints relative to their parents, the length of the bones and the position and orientation of the bones. We propose a two step framework to automatically animate a non T-pose human model into a neutral pose. First, we calculate the final position of the skeleton joints in order to form a T-pose skeleton. Suppose the relative initial length and normal direction of joint j to its parent is given by l and $\mathbf{d} = [u_1 \; v_1 \; w_1]$ and the actual direction of the same joint to its parents for a normalized T-pose skeleton is given by $\mathbf{d}_t = [u_T \; v_T \; w_T]$. The final position of joint j relative to its parent for a normalized T-pose skeleton is then given by Eq. 5.15.

$$\mathbf{p}_T = l \times \mathbf{d}_T \tag{5.15}$$

Note that the bone directions for a T-pose skeleton is fixed for human models and once obtained they can be used for different skeletons and models. Also, the length and direction of each joint are relative to their parents.

After we get the initial frame, which is the initial position of the model, and the final frame, which is the model transferred into a T-pose, we can interpolate them to create some intermediate frames. For this research we used linear interpolation to get the joints positions of the intermediate frames. However, one can use other interpolation approaches to get more robust results. Suppose we would like to have an animation clip with n frames. The position of joint j in frame k is given in Eq. 5.16, where $\mathbf{p}_0$ is the position of joint j in the first frame, with initial position.

$$\mathbf{p}_k = \frac{k \times (\mathbf{p}_0 + \mathbf{p}_T)}{n-1} \tag{5.16}$$

We should also maintain the physical property of human body movements, for example the maximum and minimum angles that each joint can have through rotation. For each frame we check these properties to make sure that they are on the natural range and the limits are the Degree of Freedom (DOF) for each joint which can be obtained from a MoCap file. Suppose each joint j can rotate along the three axis with the values of $\mathbf{r}_j$. Then to maintain the physical property of human movement we will have Eq. 5.17, where $\mathbf{min}_j$ and $\mathbf{max}_j$ corresponds the minimum and maximum of the DOF respectively.

$$\mathbf{min}_j \leq \mathbf{r}_j \leq \mathbf{max}_j \tag{5.17}$$

The goal of this research is to transfer a non T-pose 3D model into a T-pose automatically and smoothly. To do so we proposed a two step framework where we initially compute the final rotation of the skeleton joints to form a T-pose skeleton and then generate some intermediate frames by linear interpolation

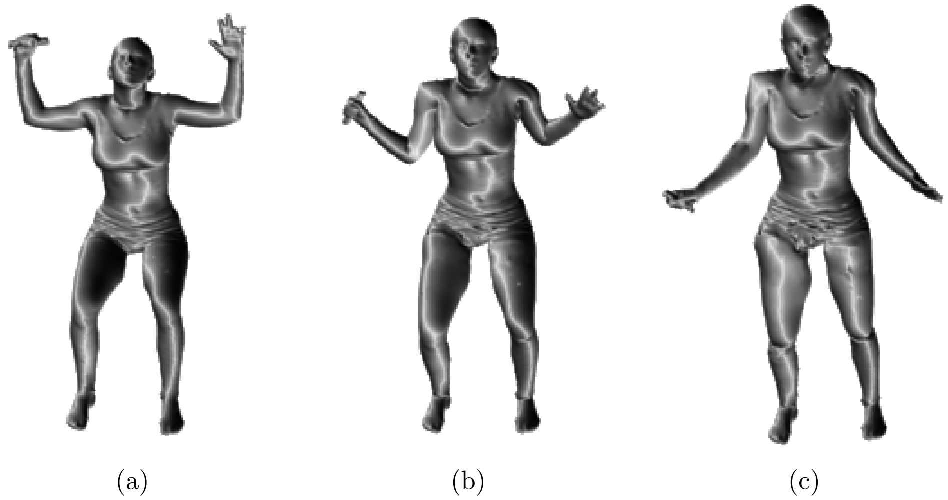


Figure 5.11: (a) The initial frame with sitting position; (b) Frame twenty shows the intermediate step (c) Final frame shows the neutral position

between the joints initial position and their final rotations. Fig. 5.11 shows the initial, one intermediate and the final frame, generated by the proposed algorithm to transfer a sitting model into a neutral pose. We created the animation through a forty step interpolation, which resulted in a forty-frame clip. The video of the generated motion is provided as supplementary material.

5.6 Experimental Results and Comparisons

The fully automatic methods that have been proposed in the literature thus far require that the models be in a front facing T position to allow embedding or extracting the animation skeleton. Other methods require input from the user, for instance selecting the head or the pelvis of the model. Our method can automatically extract the animation skeleton for an arbitrary model, in any position and orientation.

We tested our algorithm on 20 human models in different poses and gestures. The models are available in [23]. Our approach can successfully extract the animation skeleton under the assumption that the extracted curve skeleton does not contain any loop. For better visualization all of the skeletons are superimposed inside the 3D meshes. Fig. 5.12 shows how some of the joints have been relocated so that the final bones are inside the 3D model. The joints inside the circle have been relocated so that the *Correctness* property remains valid.

Fig. 5.13 shows the improvements in automated animation resulting from the optimized skeletons detected by our algorithm. Without optimization the right arm appears warped and twisted (top row); while our skeleton optimization results in a much more natural looking animated right arm. The videos of these animations are in the supplementary material, where we show that our implementation can handle models in non-T position. Animation produced by automatic skeleton embedding is also shown for comparison. Details of our implementation are not included because of limited space.

Fig. 5.14 shows some walking animation frames using the proposed method on three different models with arbitrary initial position and orientation. The proposed method extracted the animation skeleton dirctly from the model reagrdless of its initial position and orientation, therefore the rigged model can first transfer from arbitrary initial position to a neutral position and then start the motion. However, as you can see in the Fig. 5.16 the method proposed in [13] assumes that the models are in standard T-pose and front facing to the camera and animates the models based on this assumptions which creates unrealistic results. For more animation sequences on more models please refer to the supplemental material uploaded..

Fig. 5.15 compare the optimized an non-optimized extracted skeleton with [13]. It shows that [13] has a huge assumption about initial position to embed the skeleton.

Finally, Fig. 5.17 shows more examples of the proposed method on 3D models with different initial positions. We showed the models in the front facing

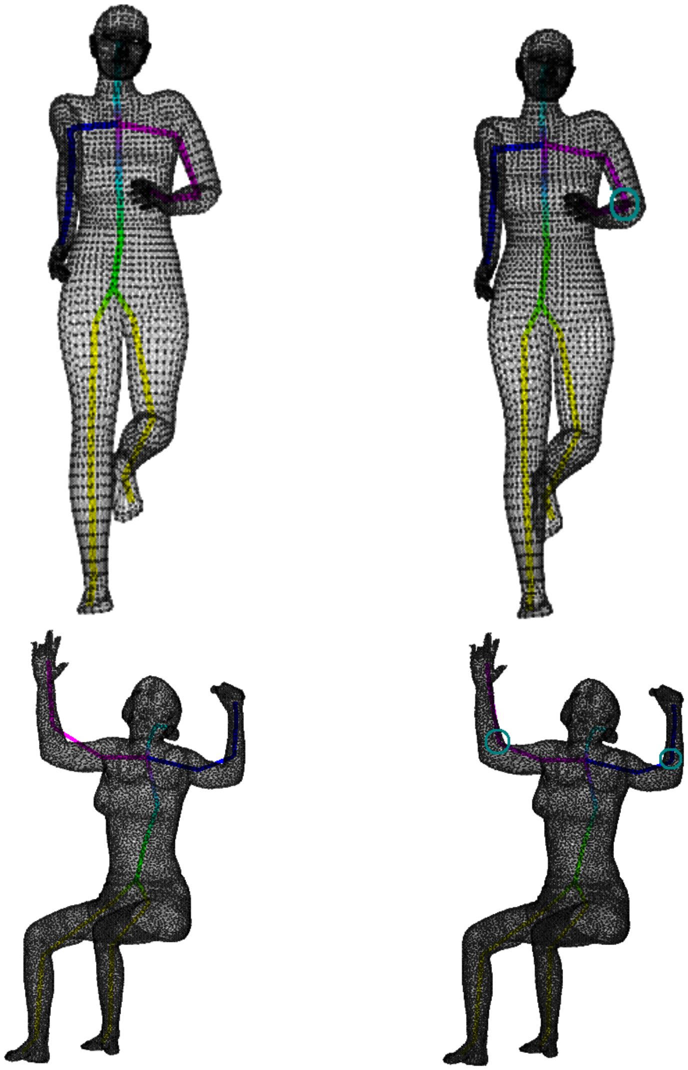


Figure 5.12: First column is the animation skeleton based on $\varphi_{Compatibility}$ property only and second column is the animation skeleton with optimization incorporating the $\varphi_{Correctness}$ property. All the skeletons are superimposed inside the meshes for better visualization.

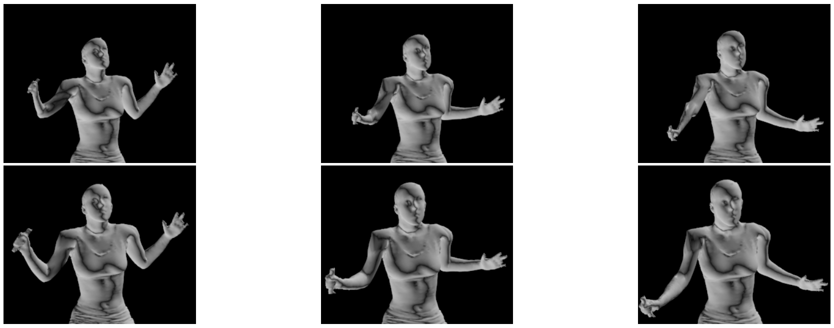


Figure 5.13: Top row: frames from animation using skeleton without optimization. Bottom row: corresponding frames using skeleton optimized by our algorithm. Note that the right arm does not look warped in the bottom row.

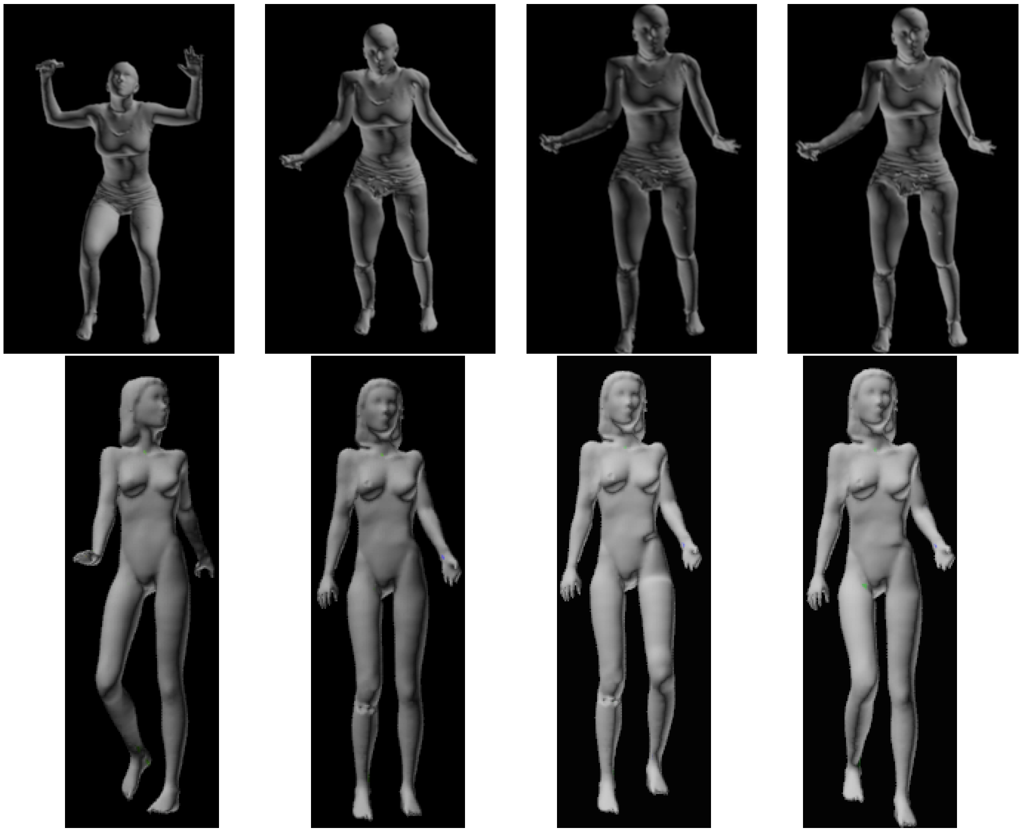


Figure 5.14: First column: initial position. Second column: transition to a neutral pose. Third and Fourth column: different frames of a walking sequence.

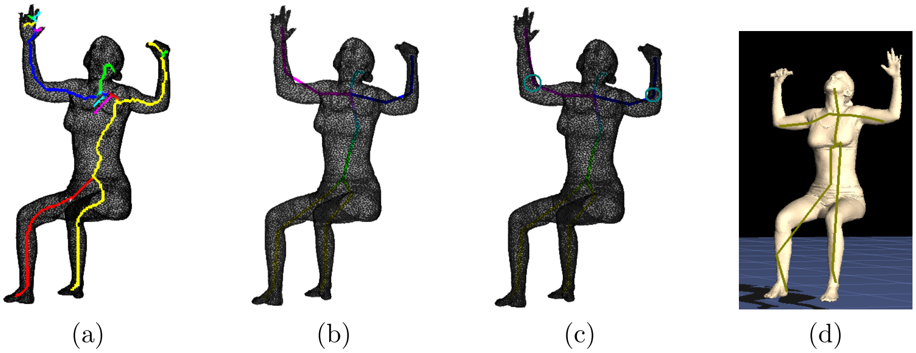


Figure 5.15: (a) The initial extracted curve skeleton(b) The animation skeleton based on $\varphi_{Compatibility}$ property only; (b) The animation skeleton with optimization incorporating the $\varphi_{Correctness}$ property; (d) The embedded skeleton based on Pinnocchio technique [13].

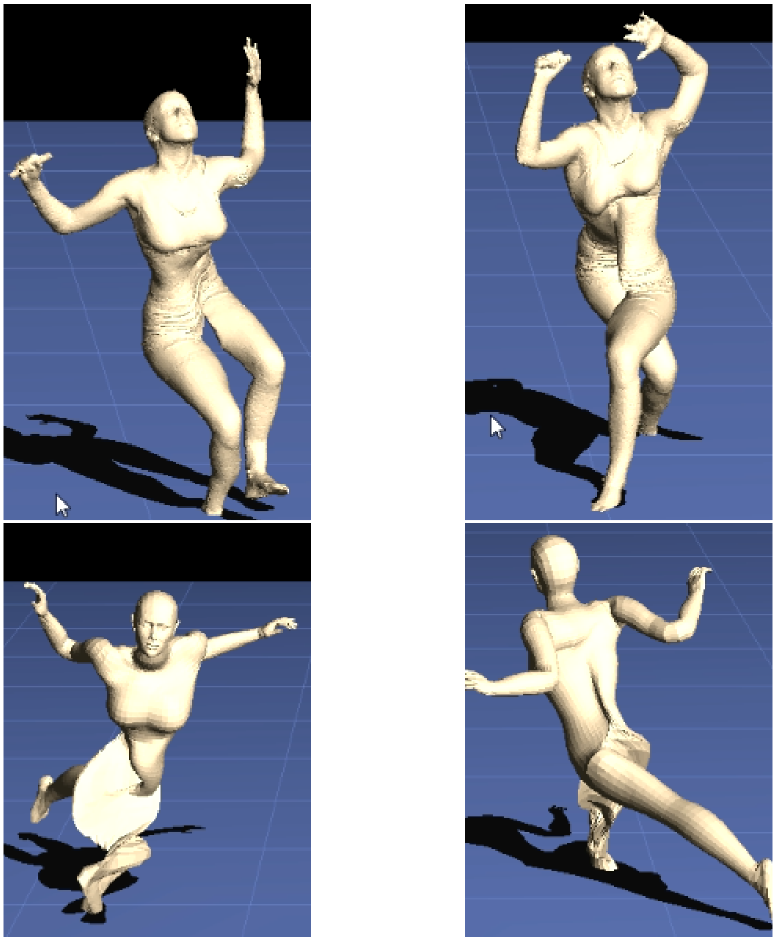


Figure 5.16: Different frames of the Pinocchio proposed by [13], on two models with arbitrary initial position and orientation.

position so that the reader can better visualize them, however the models were not originally in the front facing positions and the proposed method could successfully extract the animation skeleton. Also, note that the proposed method optimizes the curve skeleton points to find the best position for the joints. However, if the curve skeleton itself is not in the best position, the position of some joints might not look realistic; e.g., right shoulder in Fig. 5.17(i). To solve this problem one might consider optimizing the points inside a sphere around each candidate point.

5.7 Conclusion

We proposed a fully automatic approach to extract the animation skeleton directly from a 3D model. Our approach does not require the model to be in a neutral resting pose and it can accurately detect the skeleton joints as long as the extracted curve skeleton does not contain any loop. Experimental results demonstrate that our optimized animation skeleton reduces artefacts during automated animations. In future work we will study the effect of better skeletons on the quality of animation generated after MoCap compression and transmission.

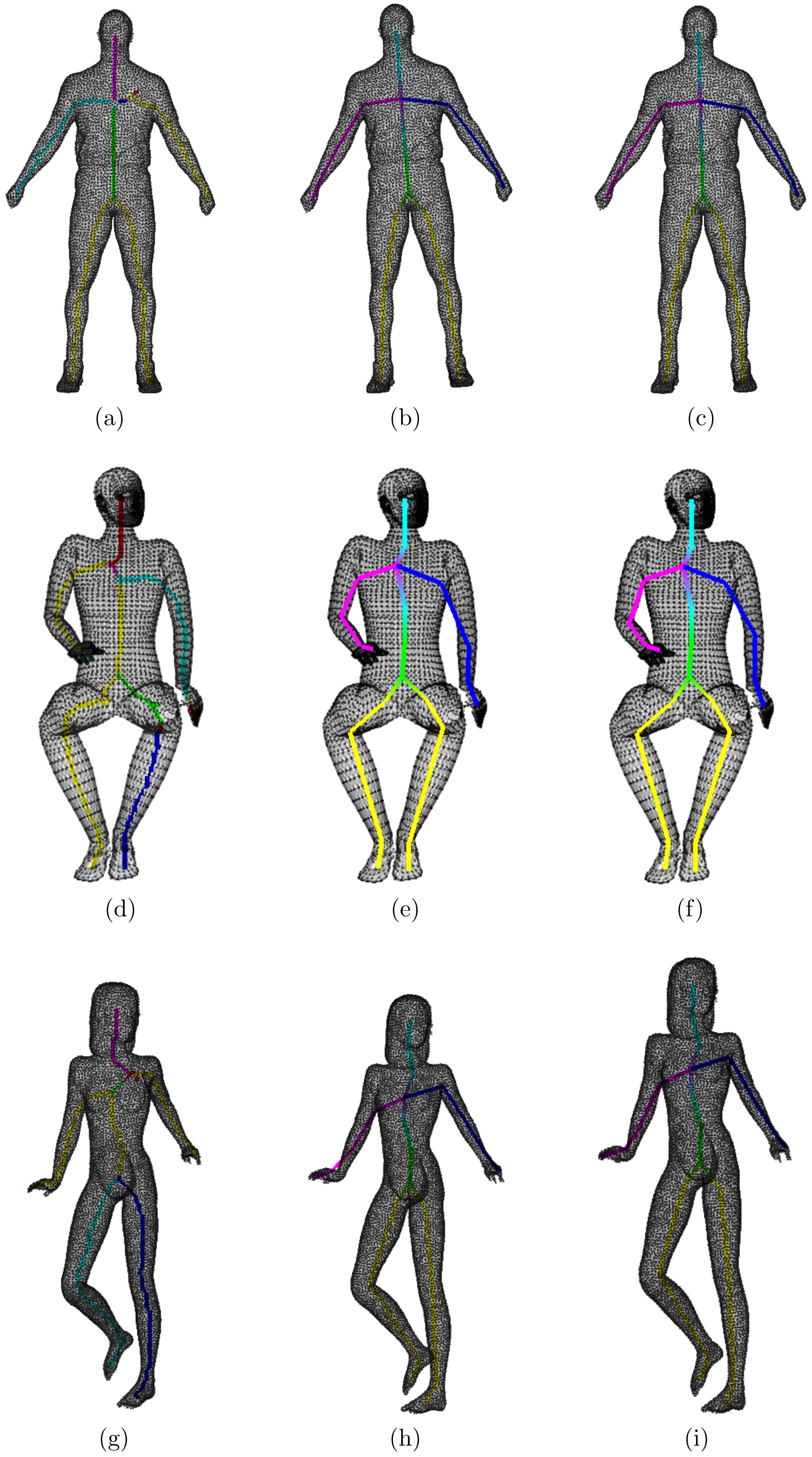


Figure 5.17: First column is the curve skeleton, second column is the animation skeleton based on *Compatibility* property only and third column is the animation skeleton with optimization. All the skeletons are superimposed inside the meshes.

Chapter 6

Conclusion and Future Direction

In this thesis we studied three different motion analysis applications in three different domains. The first application is analysing 1D eye motion to better understand team cognition between two surgeons.The second application is fall detection in 2D videos. Finally the last application is detecting the 3D trajectory key-points of arbitrary 3D human models automatically.

1. Understanding team cognition in healthcare environments by analysing surgeons' eye-gaze data is a new area of research. Although team cognition is believed to be the foundation for team performance, there is no direct and objective way to measure it, especially in the healthcare settings. In fact, the deficiency in tools for objective team assessment has been a major barrier in promoting surgical team training. In this thesis we analysed both spatial and temporal features of the eye-gaze data. The results showed that the top performance team, which is an indication of better team cognition, displayed a higher recurrence rate, lower delay, higher correlation values and higher overlap. Based on the results of our study, dual eye-tracking and CRP/CRA is demonstrated to be a powerful tool for revealing team cognition, and can help improve the training quality of a surgical team.

This study sorted the teams based on completion time. However, the design of the study should form teams based on their expertise and include expert surgeons in elite teams and inexperienced surgeons into novice teams and compare these two groups based on different factors. One could perform a new study by including surgeons with different level of surgical expertise.

2. Falling down is a dangerous incident and life threatening for seniors, especially if the person is alone and cannot seek immediate help. Therefore, automatic, real-time fall detection techniques can improve the quality of life for seniors along their independence. We proposed a real-time automatic fall detection system based on motion deformation and velocity of the action. We found that fall detection at the frame level is more desirable than fall detection at the action level. Finding starting point of an action or breaking a long video into smaller action is an ambiguous task. The proposed method detects the region of interest, which is the person, through a novel and robust ROI detection technique. It also detects two significant points, head and center of the person. The angle, size and velocity of the vector, generated by two significant points, are used to detect falling. The results show that the proposed system is quite accurate and very robust.

   In future, one could use depth cameras to address the issues of falling alongside the viewing direction of the camera. Also, it is possible to use an outlier detection approach such as One Class SVM (OCSVM) to detect the fall as an outlier action.

3. To animate an articulated 3D character an animation skeleton needs to be extracted from or be embedded into a 3D model so that the model can be automatically deformed during animation. In conventional ani-

mation software, this process is mostly done manually by expert animators, which makes it a very tedious and time-consuming step. Recently, some automatic rigging approaches have been proposed in the literature. However, most of these techniques are not fully automatic and require a front facing model in a neutral T-pose to accurately extract or embed the animation skeleton. We proposed a fully automatic approach to extract the animation skeleton directly from a 3D model. Our approach does not require the model to be in a neutral resting pose and it can accurately detect the skeleton joints as long as the extracted curve skeleton does not contain any closed loops. Experimental results demonstrate that our optimized animation skeleton reduces artefacts during automated animations.

In future, we will introduce a new system that can extract animation skeleton for any arbitrary 3D model.

Finally this three applications can be combined together to solve many healthcare and surveillance related problems.

References

[1] http://www.animalcorner.co.uk/wildlife/giraffes/giraffe_anatomy.html. 59

[2] A. A. Golovinskiy and T. Funkhouser, “Randomized cuts for 3d mesh analysis,” *ACM Trans. Graph.*, vol. 27, no. 5, 145:1–145:12, Dec. 2008. 57, 62

[3] M. S. Alzahrani, S. K. Jarraya, M. A. Salamah, and H. Ben-Abdallah, “Fallfree: Multiple fall scenario dataset of cane users for monitoring applications using kinect,” in *2017 13th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, Dec. 2017, pp. 327–333. 31

[4] M. Andriluka, S. Roth, and B. Schiele, “Pictorial structures revisited: People detection and articulated pose estimation,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 1014–1021. 30

[5] O. Arikan and D. A. Forsyth, “Interactive motion generation from examples,” in *ACM Transactions on Graphics (TOG)*, vol. 21, 2002, pp. 483–490. 73, 93

[6] M. S. Atkins, G. Tien, R. S. Khan, A. Meneghetti, and B. Zheng, “What do surgeons see capturing and synchronizing eye gaze for surgery applications,” *Surgical innovation*, vol. 20, no. 3, pp. 241–248, 2013. 11

[7] M. Attene, B. Falcidieno, and M. Spagnuolo, “Hierarchical mesh segmentation based on fitting primitives,” *Vis. Comput.*, vol. 22, no. 3, pp. 181–193, Mar. 2006. 56

[8] M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, and A. Tal, “Mesh segmentation - a comparative study,” in *IEEE International Conference on Shape Modeling and Applications*, ser. SMI ’06, IEEE Computer Society, 2006, pp. 7–. 53

[9] O. K.-C. Au, C.-L. Tai, H.-K. Chu, D. Cohen-Or, and T.-Y. Lee, “Skeleton extraction by mesh contraction,” *ACM Trans. Graph.*, vol. 27, no. 3, 44:1–44:10, Aug. 2008. 78

[10] K. M. Augestad, R.-O. Lindsetmo, H. Reynolds, J. Stulberg, A. Senagore, B. Champagne, A. G. Heriot, F. Leblanc, and C. P. Delaney, “International trends in surgical treatment of rectal cancer,” *The American Journal of Surgery*, vol. 201, no. 3, pp. 353–358, 2011. 3

[11] G. Aujay, F. Hétroy, F. Lazarus, and C. Depraz, "Harmonic skeleton for realistic character animation," in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2007, pp. 151–160. 70

[12] G. Baciu and B. K. Iu, "Motion retargeting in the presence of topological variations," *Computer Animation and Virtual Worlds*, vol. 17, no. 1, pp. 41–57, 2006. 72

[13] I. Baran and J. Popović, "Automatic rigging and animation of 3d characters," in *ACM SIGGRAPH 2007 papers*, 2007. 6, 66–68, 96, 99

[14] ——, "Automatic rigging and animation of 3d characters," in *ACM SIGGRAPH 2007 Papers*, ser. SIGGRAPH '07, 2007. 90

[15] H. Benhabiles, J.-P. Vandeborre, G. Lavoue, and M. Daoudi, "A framework for the objective evaluation of 3d models using a ground truth of human segmented 3d models," in *IEEE Shape Modeling and Applications*, 2009. 54, 55

[16] R. Berguer, D. Forkey, and W. Smith, "Ergonomic problems associated with laparoscopic surgery," *Surgical Endoscopy*, vol. 13, no. 5, pp. 466–468, 1999. 4, 9

[17] Z. P. Bian, J. Hou, L. P. Chau, and N. Magnenat-Thalmann, "Fall detection based on body part tracking using a depth camera," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 2, pp. 430–439, Mar. 2015, ISSN: 2168-2194. 31, 32

[18] G. Carone and D. Costello, "Can europe afford to grow old?," vol. 43, Sep. 2006. 4, 28

[19] M. A. Cassera, B. Zheng, D. V. Martinec, C. M. Dunst, and L. L. Swanström, "Surgical time independently affected by surgical team size," *The American journal of surgery*, vol. 198, no. 2, pp. 216–222, 2009. 3, 9

[20] I. Charfi, J. Miteran, J. Dubois, M. Atri, and R. Tourki, "Optimized spatio-temporal descriptors for real-time fall detection: Comparison of support vector machine and adaboost-based classification," *Journal of Electronic Imaging*, vol. 22, pp. 22–18, 2013. 47, 48, 50

[21] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy, "Wearable sensors for reliable fall detection," Jan. 2005, pp. 3551–3554. 28

[22] X. Chen, A. Golovinskiy, and T. Funkhouser, "A benchmark for 3d mesh segmentation," *ACM Trans. Graph.*, vol. 28, 73:1–73:12, 2009. 54, 55, 59, 62

[23] X. Chen, A. Golovinskiy, and T. Funkhouser, "A benchmark for 3d mesh segmentation," *ACM Trans. Graph.*, vol. 28, no. 3, 73:1–73:12, Jul. 2009. 79, 95

[24] J.-L. Chua, Y. C. Chang, and W. K. Lim, “A simple vision-based fall detection technique for indoor video surveillance,” *Signal, Image and Video Processing*, vol. 9, no. 3, pp. 623–633, Mar. 2015, ISSN: 1863-1711. 31, 39

[25] M. I. Coco and R. Dale, “Cross-recurrence quantification analysis of categorical and continuous time series: An r package,” *Frontiers in psychology*, vol. 5, 2014. 13

[26] N. D. Cornea, D. Silver, and P. Min, “Curve-skeleton properties, applications, and algorithms,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 3, pp. 530–548, May 2007. 69, 78

[27] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, Jun. 2005, 886–893 vol.1. 30, 37, 38

[28] P. Dollar, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: An evaluation of the state of the art,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, Apr. 2012, ISSN: 0162-8828. 37

[29] R. O. Duda and P. E. Hart, “Use of the hough transformation to detect lines and curves in pictures,” *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972, ISSN: 0001-0782. 42

[30] B. Dunkin, G. Adrales, K. Apelgren, and J. Mellinger, “Surgical simulation: A current review,” *Surgical endoscopy*, vol. 21, no. 3, pp. 357–366, 2007. 4, 10

[31] L. S. Feldman, V. Sherman, and G. M. Fried, “Using simulators to assess laparoscopic competence: Ready for widespread use?” *Surgery*, vol. 135, no. 1, pp. 28–42, 2004. 4, 10

[32] Q. Feng, C. Gao, L. Wang, M. Zhang, L. Du, and S. Qin, “Fall detection based on motion history image and histogram of oriented gradient feature,” in *2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, Nov. 2017, pp. 341–346. 32

[33] R. Flin and N. Maran, “Identifying and training non-technical skills for teams in acute medicine,” *Quality and Safety in Health care*, vol. 13, no. suppl 1, pp. i80–i84, 2004. 11

[34] A. G. Gallagher, M. Al-Akash, N. E. Seymour, and R. M. Satava, “An ergonomic analysis of the effects of camera rotation on laparoscopic performance,” *Surgical endoscopy*, vol. 23, no. 12, pp. 2684–2691, 2009. 3, 4, 9

[35] Y. M. Galvão, V. A. Albuquerque, B. J. T. Fernandes, and M. J. S. Valença, “Anomaly detection in smart houses: Monitoring elderly daily behavior for fall detecting,” in *2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, Nov. 2017, pp. 1–6. 31–33

[36] C. Ge, I. Y. H. Gu, and J. Yang, “Human fall detection using segment-level cnn features and sparse dictionary learning,” in *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, Sep. 2017, pp. 1–6. 32

[37] M. Gleicher, “Retargetting motion to new characters,” in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniq*, 1998, pp. 33–42. 66, 71, 92

[38] N. Hajari, I. Cheng, A. Basu, and G. Lavoué, “Evaluation of 3d model segmentation techniques based on animal anatomy,” in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, Oct. 2013, pp. 3277–3281. 85

[39] L. A. Haveran, Y. W. Novitsky, D. R. Czerniach, G. K. Kaban, M. Taylor, K. Gallagher-Dorval, R. Schmidt, J. J. Kelly, and D. E. Litwin, “Optimizing laparoscopic task efficiency: The role of camera and monitor positions,” *Surgical endoscopy*, vol. 21, no. 6, pp. 980–984, 2007. 3, 9

[40] M. M. Hayhoe, “Advances in relating eye movements and cognition,” *Infancy*, vol. 6, no. 2, pp. 267–274, 2004. 8

[41] A. Healey, S. Undre, and C. Vincent, “Developing observational measures of performance in surgical teams,” *Quality and Safety in Health Care*, vol. 13, no. suppl 1, pp. i33–i40, 2004. 4, 10

[42] C. Hecker, B. Raabe, R. W. Enslow, J. DeWeese, J. Maynard, and K. van Prooijen, “Real-time motion retargeting to highly varied user-created morphologies,” in *ACM Transactions on Graphics (TOG)*, vol. 27, 2008, p. 27. 72

[43] E. S. L. Ho, H. P. H. Shum, Y. ming Cheung, and P. C. Yuen, “Topology aware data-driven inverse kinematics.,” *Comput. Graph. Forum*, vol. 32, no. 7, pp. 61–70, 2013. 66

[44] J. Hu, R. Kashi, D. Lopresti, G. Nagy, and G. Wilfong, “Why table ground-truthing is hard,” in *International Conference on Document Analysis and Recognition*, Seattle, Washington, USA, 2001. 55

[45] R. Igual, C. Medrano, and I. Plaza, “Challenges, issues and trends in fall detection systems,” *BioMedical Engineering OnLine*, vol. 12, no. 1, p. 66, Jul. 2013, ISSN: 1475-925X. 5, 28

[46] D. Jacka, A. Reid, B. Merry, and J. Gain, “A comparison of linear skinning techniques for character animation,” in *Proceedings of the 5th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, 2007, pp. 177–186. 66

[47] A. Jacobson, I. Baran, L. Kavan, J. Popović, and O. Sorkine, “Fast automatic skinning transformations,” *ACM Trans. Graph.*, vol. 31, no. 4, 77:1–77:10, Jul. 2012. 66

[48] P. KaewTraKulPong and R. Bowden, “An improved adaptive background mixture model for real-time tracking with shadow detection,” in *Video-Based Surveillance Systems: Computer Vision and Distributed Processing*, P. Remagnino, G. A. Jones, N. Paragios, and C. S. Regazzoni, Eds. Boston, MA: Springer US, 2002, pp. 135–144. 36

[49] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri, “3d shape segmentation with projective convolutional networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 6630–6639. 87

[50] E. Kalogerakis, A. Hertzmann, and K. Singh, “Learning 3d mesh segmentation and labeling,” *ACM Trans. Graph.*, vol. 29, no. 4, 102:1–102:12, Jul. 2010. 85, 87, 88, 90

[51] J. M. Kamal Sehairi Fatima Chouireb, “Comparative study of motion detection methods for video surveillance systems,” *Journal of Electronic Imaging*, vol. 26, pp. 26–29, 2017. 34

[52] S. Katz, G. Leifman, and A. Tal, “Mesh segmentation using feature point and core extraction,” *The Visual Computer*, vol. 21, no. 8-10, pp. 649–658, 2005. 56

[53] S. Katz and A. Tal, “Hierarchical mesh decomposition using fuzzy clustering and cuts,” *ACM Trans. Graph.*, vol. 22, no. 3, pp. 954–961, Jul. 2003. 56

[54] L. Kavan, S. Collins, J. Žára, and C. O’Sullivan, “Skinning with dual quaternions,” in *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, 2007, pp. 39–46. 74

[55] L. Kavan and J. Žára, “Spherical blend skinning: A real-time deformation of articulated models,” in *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games*, 2005, pp. 9–16. 74

[56] T. A. Kenyon, M. P. Lenker, T. Bax, and L. Swanstrom, “Cost and benefit of the trained laparoscopic team,” *Surgical endoscopy*, vol. 11, no. 8, pp. 812–814, 1997. 4, 9

[57] R. S. Khan, G. Tien, M. S. Atkins, B. Zheng, O. N. Panton, and A. T. Meneghetti, “Analysis of eye gaze: Do novice surgeons look at the same location as expert surgeons during a laparoscopic operation?” *Surgical endoscopy*, vol. 26, no. 12, pp. 3536–3540, 2012. 11, 17

[58] L. Klack, C. Möllering, M. Ziefle, and T. Schmitz-Rode, “Future care floor: A sensitive floor for movement monitoring and fall detection in home environments,” in *Wireless Mobile Communication and Healthcare*, J. C. Lin and K. S. Nikita, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 211–218, ISBN: 978-3-642-20865-2. 29

[59] L. Kovar, M. Gleicher, and F. Pighin, “Motion graphs,” in *ACM transactions on graphics (TOG)*, vol. 21, 2002, pp. 473–482. 73, 93

[60] J. Kwon and F. C. Park, “Natural movement generation using hidden markov models and principal components,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 38, no. 5, pp. 1184–1194, 2008. 73, 93

[61] D. Lahiri, C. Dhiman, and D. K. Vishwakarma, “Abnormal human action recognition using average energy images,” in *2017 Conference on Information and Communication Technology (CICT)*, Nov. 2017, pp. 1–5. 31

[62] Y.-K. Lai, S.-M. Hu, R. Martin, and P. Rosin, “Fast mesh segmentation using random walks,” in *ACM symposium on Solid and physical modeling*, Stony Brook, New York, 2008, pp. 183–191. 57, 62

[63] J. Lee, J. Chai, P. S. Reitsma, J. K. Hodgins, and N. S. Pollard, “Interactive control of avatars animated with human motion data,” in *ACM Transactions on Graphics (TOG)*, vol. 21, 2002, pp. 491–500. 73, 93

[64] X. Li, T. Pang, W. Liu, and T. Wang, “Fall detection for elderly person care using convolutional neural networks,” in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Oct. 2017, pp. 1–6. 32

[65] W. N. Lie, A. T. Le, and G. H. Lin, “Human fall-down event detection based on 2d skeletons and deep learning approach,” in *2018 International Workshop on Advanced Image Technology (IWAIT)*, Jan. 2018, pp. 1–4. 32

[66] H. Liu, Y. Qian, and S. Lin, “Detecting persons using hough circle transform in surveillance video.,” in *VISAPP (2)*, 2010, pp. 267–270, ISBN: 978-989-674-029-0. 42

[67] P.-C. Liu, F.-C. Wu, W.-C. Ma, R.-H. Liang, and M. Ouhyoung, “Automatic animation skeleton construction using repulsive force field,” in *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, 2003, pp. 409–. 69

[68] W. Lu, Y. Liu, J. Sun, and L. Sun, “A motion retargeting method for topologically different characters,” in *Computer Graphics, Imaging and Visualization, 2009. CGIV’09. Sixth International Conference on*, 2009, pp. 96–100. 72

[69] J. Lv, X. Chen, J. Huang, and H. Bao, “Semi-supervised mesh segmentation and labeling,” *Computer Graphics Forum*, vol. 31, no. 7, pp. 2241–2248, 85

[70] A. D. Malkan, A. H. Loh, and J. A. Sandoval, “Minimally invasive surgery in the management of abdominal tumors in children,” *Journal of pediatric surgery*, vol. 49, no. 7, pp. 1171–1176, 2014. 3

[71] M. J. Marin-Jimenez, A. Zisserman, M. Eichner, and V. Ferrari, "Detecting people looking at each other in videos," *International Journal of Computer Vision*, vol. 106, no. 3, pp. 282–296, Feb. 2014, ISSN: 1573-1405. 42

[72] D. Martin, C. Fowlkes, D. Tal, and Jitendra Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *International Conference on Computer Vision*, 2001. 55

[73] N. Marwan, M. C. Romano, M. Thiel, and J. Kurths, "Recurrence plots for the analysis of complex systems," *Physics Reports*, vol. 438, no. 5, pp. 237–329, 2007. 12, 13

[74] T. McCracken, R. Kainer, and D. Carlson, *Color Atlas of Small Animal Anatomy the essentials*. Iowa State University Press, 2009. 59, 60

[75] T. McCracken, R. Kainer, T. Spurgeon, and G. Brooks, *Spurgeon's Color Atlas of Large Animal Anatomy: The Essentials*. John Wiley and Sons Ltd, 2008. 59, 60

[76] F. Merrouche and N. Baha, "Fall detection using head tracking and centroid movement based on a depth camera," in *Proceedings of the International Conference on Computing for Engineering and Sciences*, ser. ICCES '17, Istanbul, Turkey, 2017, pp. 29–34. 31–33, 39

[77] B. Mirmahboub, S. Samavi, N. Karimi, and S. Shirani, "Automatic monocular system for human fall detection based on variations in silhouette area," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 2, pp. 427–436, Feb. 2013, ISSN: 0018-9294. 31–33, 39

[78] T. Möller and B. Trumbore, "Fast, minimum storage ray-triangle intersection," *J. Graph. Tools*, vol. 2, no. 1, pp. 21–28, Oct. 1997. 83

[79] J.-S. Monzani, P. Baerlocher, R. Boulic, and D. Thalmann, "Using an intermediate skeleton and inverse kinematics for motion retargeting," *Computer Graphics Forum*, 2000. 72

[80] K. Moorthy, Y. Munz, S. Adams, V. Pandey, and A. Darzi, "A human factors analysis of technical and team skills among surgical trainees during procedural simulations in a simulated operating theatre," *Annals of surgery*, vol. 242, no. 5, p. 631, 2005. 4, 14

[81] M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, pp. 144–152, 2013, Special issue: Behaviours in video, ISSN: 0925-2312. 28, 29

[82] R. S. Nickerson, S. T. Dumais, S. Lewandowsky, T. J. Perfect, and F. T. Durso, *Handbook of applied cognition*. John Wiley & Sons, 2007. 3

[83] N. Noury, P. Rumeau, A. Bourke, G. ÓLaighin, and J. Lundy, "A proposal for the classification and evaluation of fall detectors," *IRBM*, vol. 29, no. 6, pp. 340–349, 2008, ISSN: 1959-0318. 43

[84] J. Pan, X. Yang, X. Xie, P. Willis, and J. J. Zhang, "Automatic rigging for animation characters with 3d silhouette," *Comput. Animat. Virtual Worlds*, vol. 20, no. 23, pp. 121–131, Jun. 2009. 70

[85] N. Pantuwong and M. Sugimoto, "A novel template-based automatic rigging algorithm for articulated-character animation," *Journal of Visualization and Computer Animation*, vol. 23, no. 2, pp. 125–141, Mar. 2012. 70

[86] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers, "A review of wearable sensors and systems with application in rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 9, no. 1, p. 21, Apr. 2012, ISSN: 1743-0003. 28

[87] A. Pietak, S. Ma, C. W. Beck, and M. D. Stringer, "Fundamental ratios and logarithmic periodicity in human limb bones.," *Journal of Anatomy*, 2013. 80

[88] M. Poirier and E. Paquette, "Rig retargeting for 3d animation," in *Proceedings of Graphics Interface 2009*, 2009, pp. 103–110. 68

[89] M. Prastawa, E. Bullitt, and G. Gerig, "Synthetic ground truth for validation of brain tumor mri segmentation," in *MICCAI*, Palm Spring, USA, 2005, pp. 26–33. 55

[90] D. C. Richardson and R. Dale, "Looking to understand: The coupling between speakers' and listeners' eye movements and its relationship to discourse comprehension," *Cognitive science*, vol. 29, no. 6, pp. 1045–1060, 2005. 12, 19

[91] L. Richstone, M. J. Schwartz, C. Seideman, J. Cadeddu, S. Marshall, and L. R. Kavoussi, "Eye metrics as an objective assessment of surgical skill," *Annals of surgery*, vol. 252, no. 1, pp. 177–182, 2010. 3, 42

[92] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Monocular 3d head tracking to detect falls of elderly people," in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug. 2006, pp. 6384–6387. 33, 39, 40, 42

[93] ——, "Robust video surveillance for fall detection based on human shape deformation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 611–622, May 2011, ISSN: 1051-8215. 30, 33, 39

[94] E. Salas, K. A. Wilson, C. S. Burke, and H. A. Priest, "Using simulation-based training to improve patient safety: What does it take?" *Joint Commission Journal on Quality and Patient Safety*, vol. 31, no. 7, pp. 363–371, 2005. 4, 10

[95] A. Savenko and G. Clapworthy, "Using motion analysis techniques for motion retargeting," in *Information Visualisation, 2002. Proceedings. Sixth International Conference on*, 2002, pp. 110–115. 72, 92

[96] S. Schaefer and C. Yuksel, “Example-based skeleton extraction,” in *Proceedings of the fifth Eurographics symposium on Geometry processing*, 2007, pp. 153–162. 70

[97] K. Sehairi, F. Chouireb, and J. Meunier, “Elderly fall detection system based on multiple shape features and motion analysis,” in *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*, Apr. 2018, pp. 1–8. 31, 33, 37, 39, 50

[98] A. Shamir, “A survey on mesh segmentation techniques,” *Computer Graphics Forum*, vol. 27, no. 6, pp. 1539–1556, 2008. 87

[99] L. Shapira, A. Shamir, and D. Cohen-Or, “Consistent mesh partitioning and skeletonisation using the shape diameter function,” *Vis. Comput.*, vol. 24, no. 4, pp. 249–259, Mar. 2008. 57, 62

[100] L. Shi, I. Cheng, and A. Basu, “Anatomy preserving 3d model decomposition based on robust skeleton-surface node correspondence,” in *IEEE International Conference on Multimedia and Expo*, 2011, pp. 1–6. 54, 58, 59, 62

[101] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, “The princeton shape benchmark,” in *Shape Modeling International*, 2004, pp. 167–178. 59

[102] E. M. Shore, G. G. Lefebvre, and T. P. Grantcharov, “Gynecology resident laparoscopy training: Present and future,” *American journal of obstetrics and gynecology*, vol. 212, no. 3, pp. 298–301, 2015. 3

[103] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2, Fort Collins, CO, USA, Aug. 1999, 246–252 Vol. 2. 34, 38

[104] S. Sukumar, M. Sun, P. I. Karakiewicz, A. A. Friedman, F. K. Chun, J. Sammon, K. R. Ghani, P. Ravi, M. Bianchi, W. Jeong, *et al.*, “National trends and disparities in the use of minimally invasive adult pyeloplasty,” *The Journal of urology*, vol. 188, no. 3, pp. 913–918, 2012. 3

[105] R. W. Sumner and J. Popović, “Deformation transfer for triangle meshes,” *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 399–405, 2004. 72

[106] S. Tak and H.-S. Ko, “A physically-based motion retargeting filter,” *ACM Transactions on Graphics (TOG)*, vol. 24, no. 1, pp. 98–117, 2005. 72

[107] M. Teichmann and S. Teller, “Assisted articulation of closed polygonal models,” in *ACM SIGGRAPH 98 Conference abstracts and applications*, 1998, pp. 254–. 69

[108] G. Tien, M. S. Atkins, X. Jiang, R. Khan, and B. Zheng, “Identifying eye gaze mismatch during laparoscopic surgery.,” *Studies in health technology and informatics*, vol. 184, pp. 453–457, 2012. 11, 17

[109] G. Tien, B. Zheng, and M. S. Atkins, "Quantifying surgeons' vigilance during laparoscopic operations using eyegaze tracking.," in *MMVR*, 2011, pp. 658–662. 11

[110] Y.-Y. Tsai, H.-K. Chu, K. B. Cheng, T.-Y. Lee, and C.-L. Yen, "Animation generation and retargeting based on physics characteristics," in *Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007. Third International Conference on*, vol. 1, 2007, pp. 349–352. 73

[111] S. Undre, N. Sevdalis, A. N. Healey, S. A. Darzi, and C. A. Vincent, "Teamwork in the operating theatre: Cohesion or confusion?" *Journal of evaluation in clinical practice*, vol. 12, no. 2, pp. 182–189, 2006. 4, 10

[112] *Vicon motion capture system*, `http://www.vicon.com`. 71, 92

[113] D. Vlasic, I. Baran, W. Matusik, and J. Popović, "Articulated mesh animation from multi-view silhouettes," in *ACM SIGGRAPH 2008 papers*, Aug. 2008, 97:1–97:9. 66

[114] P. Volino, M. Courchesne, and N. Magnenat Thalmann, "Versatile and efficient techniques for simulating cloth and other deformable objects," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp. 137–144. 66

[115] L. Wade and R. E. Parent, "Fast, fully-automated generation of control skeletons for use in animation," in *Proceedings of the Computer Animation*, 2000, pp. 164–. 69

[116] T. Wang, I. Cheng, V. Lopez, E. Bribiesca, and A. Basu, "Valence normalized spatial median for skeletonization and matching," in *ICCV Workshops*, 2009, pp. 55–62. 58

[117] T. Wang, I. Cheng, V. Lopez, E. Bribiesca, and A. Basu, "Valence normalized spatial median for skeletonization and matching," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, 27 2009-Oct. 4, pp. 55–62. 79

[118] S. Warfield, K. Zou, and W. Wells, "Validation of image segmentation and expert quality with an expectation maximization algorithm," in *MICCAI*, Tokyo, Japan, 2002, pp. 298–306. 55

[119] *World health organization: Global report on falls prevention in older age*, `http://www.who.int/ageing/publications/Falls_prevention7March.pdf`. 4, 28

[120] X. Yang, A. Somasekharan, and J. J. Zhang, "Curve skeleton skinning for human and creature characters," *Computer Animation and Virtual Worlds*, vol. 17, no. 3-4, pp. 281–292, 2006. 74

[121] B. A. Yanikoglu and L. Vicent, "Pink panther: A complete environment for ground-truthing and benchmarking document page segmentation," *Pattern Recognition*, vol. 31, no. 9, pp. 1191–1204, 1998. 55

[122] L. Yao, W. Min, and K. Lu, "A new approach to fall detection based on the human torso motion model," *Applied Sciences*, vol. 7, no. 10, 2017, ISSN: 2076-3417. 31, 32

[123] M. Yu, Y. Yu, A. Rhuma, S. M. R. Naqvi, L. Wang, and J. A. Chambers, "An online one class support vector machine-based person-specific fall detection system for monitoring an elderly individual in a room environment," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 6, pp. 1002–1014, Nov. 2013, ISSN: 2168-2194. 30, 33, 39

[124] B. Zheng, P. Denk, D. Martinec, P. Gatta, M. Whiteford, and L. Swanström, "Building an efficient surgical team using a bench model simulation: Construct validity of the legacy inanimate system for endoscopic team training (lisett)," *Surgical endoscopy*, vol. 22, no. 4, pp. 930–937, 2008. 4, 10

[125] B. Zheng, Z. Janmohamed, and C. MacKenzie, "Reaction times and the decision-making process in endoscopic surgery," *Surgical Endoscopy and Other Interventional Techniques*, vol. 17, no. 9, pp. 1475–1480, 2003. 4, 9

[126] B. Zheng, L. Swanström, and C. L. Mackenzie, "A laboratory study on anticipatory movement in laparoscopic surgery: A behavioral indicator for team collaboration," *Surgical endoscopy*, vol. 21, no. 6, pp. 935–940, 2007. 11

[127] B. Zheng, F. Verjee, A. Lomax, and C. MacKenzie, "Video analysis of endoscopic cutting task performed by one versus two operators," *Surgical Endoscopy and Other Interventional Techniques*, vol. 19, no. 10, pp. 1388–1395, 2005. 10, 11

[128] B. Zheng, X. Jiang, G. Tien, A. Meneghetti, O. N. M. Panton, and M. S. Atkins, "Workload assessment of surgeons: Correlation between nasa tlx and blinks," *Surgical endoscopy*, vol. 26, no. 10, pp. 2746–2750, 2012. 11

[129] Y. Zigel, D. Litvak, and I. Gannot*, "A method for automatic fall detection of elderly people using floor vibrations and sound—proof of concept on human mimicking doll falls," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 12, pp. 2858–2867, Dec. 2009, ISSN: 0018-9294. 29