# ALGORITHMS FOR VISION BASED AMENDMENT IN HUMAN ACTION WITH ANTHROPOMETRIC INVARIANCE

A THESIS SUBMITTED TO

**SAVITRIBAI PHULE PUNE UNIVERSITY**

FOR AWARD OF DEGREE OF

**DOCTOR OF PHILOSOPHY (Ph.D)**

IN THE FACULTY OF

**COMPUTER ENGINEERING**


SUBMITTED BY

**MS. GEETANJALI PANJABRAO SABLE**

UNDER THE GUIDENCE OF

**PROF. (DR.) VARSHA H. PATIL**

RESEARCH CENTRE

**DEPARTMENT OF COMPUTER ENGINEERING**

MATOSHRI COLLEGE OF ENGINEERING & RESEARCH CENTRE

EKLAHARE, NASHIK.


**JUNE 2017**

**Matoshri College of Engineering & Research Centre**
**Eklahare, Nashik**
DEPARTMENT OF COMPUTER ENGINEERING


# Certificate

This is to certify that the work incorporated in the thesis, **"Algorithms for vision based amendment in human action with anthropometric invariance"** is submitted by **Ms. Geetanjali Panjabrao Sable** for the **Doctor of Philosophy (Ph.D)** in **Computer Engineering**, **Savitribai Phule Pune University,** has been carried out by the candidate at **Department of Computer Engineering, Matoshri College of Engineering and Research Centre Eklahare, Nashik** during the period from **March, 2013 to April, 2017** under the guidance of **Prof.** (**Dr.) Varsha Hemant Patil**.

Prof. (Dr.) Varsha H. Patil
Research Guide,
Vice Principal and Head,
Computer Engineering,
MCOERC, Eklahare, Nashik.

Prof. (Dr.) Gajanan K. Kharate
Principal,
MCOERC, Eklahare, Nashik.

Matoshri College of Engineering & Research Centre
Eklahare, Nashik
DEPARTMENT OF COMPUTER ENGINEERING

# Certificate of the Guide

Certified that the work incorporated in the thesis **"Algorithms for Vision Based Amendment in Human Action with Anthropometric Invariance"** submitted by **Ms. Geetanjali Panjabrao Sable** was carried out by the candidate for the **Doctor of Philosophy (Ph. D)** degree at **Department of Computer Engineering, Matoshri College of Engineering and Research Centre, Eklahare, Nashik** during the period from **15 March, 2013 to April, 2017** under my direct supervision and guidance. Such material has been obtained from other sources has been duly acknowledged in the thesis.

Place:
Date:

Prof. (Dr.) Varsha H. Patil
Research Guide,
Vice Principal and Head,
Computer Engineering,
MCOERC, Eklahare, Nashik.

# Declaration by the Candidate

I hereby declare that the thesis entitled **"Algorithms for vision based amendment in human action with anthropometric invariance"** submitted by me to the **Savitribai Phule Pune University, Pune** for the degree of **Doctor of Philosophy (PhD)** in **Computer Engineering,** is the record of work carried out by me during the period from **March 2013 to April 2017** under the guidance of **Prof. (Dr.) Varsha Hemant Patil** and has not formed the basis for the award of any degree, diploma, associateship, fellowship, titles in this or any other University or other institution of Higher learning.

I further declare that the material obtained from other sources has been duly acknowledged in the thesis.

Place:
Date:

Ms. Geetanjali Panjabrao Sable
Research Scholar,
Department of Computer Engineering,
MCOERC, Eklahare, Nashik.

# *Dedicated to*

*My beloved twin sons* ***Rutvij & Rushikesh***

*My Husband* ***Vinayak***

*&*

*My Parents*

***Smt. Vimal & Shri. Panjabrao Sable***

***Smt. Indumati & Late Shri. Dattatraya Kale***

# ACKNOWLEDGEMENT

I am grateful to **Lord Ganesh** for his special blessings that he has always been showering on me.

It is an honour for me to thank those who made this research work possible. My first debt of gratitude goes to my guide, worthy and esteemed mentor and advisor, Prof. (Dr.) Varsha Hemant Patil for the adroit guidance, constructive suggestions, critical evaluation and perceptual encouragement in completing my thesis. Her advice, guidance, ideas, insights and encouragement are instrumental in making this research work possible.

I render my deep sense of gratitude to respected Prof. (Dr.) Gajanan K. Kharate, Principal, MCOERC without whom this would never come to be today's reality. His constant encouragement and constructive suggestions helped me a lot to make this dream come true.

Special thanks to the committee members Prof. (Dr.) Shirish S. Sane and Dr. R. S. Tiwari for their valuable suggestions, which were instrumental in improving the quality of the research. I am also thankful to Prof. (Dr.) Aditya A. Abhyankar for his expert suggestions and encouragement.

I am also thankful to Prof. N. L. Bhale, HOD I.T., Dr. S. Pable, Research Dean, Dr. Neeta Deshpande and other staff members of MCOERC, Nashik for their suggestions and support during this research journey. My special thanks to Prof. Swati Bhavsar for all required support and motivation.

I am especially thankful to Dr. Ruta Kulkarni and other *Yogāsana* experts of *Yogapoint*, Mr. P.P. Krishnan and Mr. Sudhakar P. Nanaware *Yoga* teacher for their guidance and help for *Yogāsana* dataset creation. I would like to specially acknowledge the cooperation and efforts of the PICT students for performing *Yogāsana* and *Bharatnāṭyam* during dataset creation.

I also express my sincere thanks and deepest sense of gratitude to PICT Managing trustee Mr. R. S. Kothawale, Secretory Mr. Swastik Shisikar and Principal Dr. P. T Kulkarni, PICT, Pune. I am also thankful to Dr. Sarang Joshi, Dr. R. B. Ingle, HOD, Department of Computer Engineering, Dr. G. P. Potdar, and Dr. B. A. Sonkamble, PICT, Pune for their whole hearted cooperation and support.

Special thanks to Board of College and University Development, Savitribai Phule Pune University for financial assistance for the project under research project scheme.

I am grateful to my parents, Shri. Panjabrao Sable, Smt. Vimal Sable, Smt. Indumati Kale and all family members for their moral support and blessings. My heartful thanks go to my husband, Vinayak for his patience and understanding and my twin sons Mast. Rushikesh and Mast. Rutvij who had missed many affectionate hours during this work. I would also like to thank my younger sister Ms. Sapana, brother Mr. Ganesh and sister in law Ms. Geetanjali for their support, motivation and encouragement.

I am also thankful to my fellow colleagues Mrs. Snehal Kamalapurkar, Mrs. Chhaya Gosavi, Mrs. Archana Ghotkar, Mrs. Kalyani Waghmare, Mrs. Arati Deshpande, Mrs. Sheetal Sonawane, Mrs. Anupama Phakatkar, Ms. Manjita Kale and Mrs. Mousami Munot for their motivation, support and help thought out the journey.

I would like to take this opportunity to express my thanks to all my friends and colleagues for their full support, cooperation and help. I also thank all those who have helped me directly or indirectly in my research work.

*Geetanjali Panjabrao Sable*

# TABLE OF CONTENTS

# ABSTRACT

Vision based human motion recognition is a systematic approach to understand and analyze the movement of people in camera captured contents. Systems available for recognition of human action for various applications classify performed action in its correct class, however does not comment on incorrect actions. Some applications like traditional dances, sports, exercises require body movements to be done in a specific manner. This demands assessment of correctness of movements performed by practitioner, specifically for the novice practitioner. Corrective suggestions need to be provided for incorrect actions by an expert. But, due to unavailability or time suitability of expert and practitioner, it is not feasible for everyone to perform these actions under expert's supervision.

In this research work, algorithms for vision based amendment in human action with anthropometric invariance are designed, implemented and tested on *Yogāsana* dataset and the system is named as e-*YogaGuru*. Main motivation behind consideration of *Yogāsana* as dataset for suggestion of amendment in human action is its ability to soothe the nerves, and calm the brain to make the mind fresh and relaxed, and the body healthy and active. Viewing recorded *Yogāsana* and imitating it, may help practitioner, but it does not provide feedback about correctness of performed *Yogāsana*. Practicing *Yogāsana* incorrectly may lead to some injuries. So, the intelligent feedback system for assessment of correctness of *Yogāsana* is essential. Feedback provided by *e-YogaGuru* system helps practitioner to assess the progress and improvement in performing *Yogāsana*. Replay facility of performed *Yogāsana* helps practitioner to analyze erroneous or missing postures. It also helps expert to monitor progress of practitioner.

Main contribution of this research work is algorithms designed, implemented and tested for e-*YogaGuru* system that classifies performed *Yogāsana* in correct class. Further, erroneous *Yogāsana* is analyzed and abstract as well as detailed amendment is suggested in order to provide audio-visual feedback to practitioner. Abstract level amendment provide details of missing or erroneous posture in required posture sequence of *Yogāsana* in terms of posture name and position. Detailed amendment provides details of practitioner's body angle at erroneous postures along with required correct body angle values and video frame positions. This research work focuses on two main challenges as variability introduced due to human anthropometry and execution speed of action performed.

Experimentation of action recognition is done on *Yogāsana* using human body angles and on *Bharatnāṭyam Adavu* using human body contour. *Yogāsana* recognition results are almost 100% whereas *Bharatnāṭyam* recognition results are 78.38%. It is observed that using body angle features is advantageous as compared to body image features as they are more discriminative, robust, less sensitive to human anthropometry and provide reduced feature vector. So, *e-YogaGuru* system uses angle features for representation of motion.

State model is designed and developed for speed invariant recognition of *Yogāsana*. It also provide advantage of reduced training data. It provides flexibility of identification of incorrect posture and suggestion of amendment in terms of body angles at specific posture position with frame number for *Yogāsana* performed erroneously. Two layer hierarchical model for *Yogāsana* recognition and three layer model for suggestion of amendment is used. Dataset for five standing and two forward bending *Yogāsana,* and five *Bharatnāṭyam Adavu* is created.

e-*YogaGuru* system implemented using designed algorithms provides flexibility of time and comfort to a person to perform *Yogāsana* in absence of expert. In future, designed algorithms can also be applied for other applications like sports, different exercises and traditional dances by changing knowledge-base and state transition model.

***Keyword:***

*Activity recognition and understanding, scene anomaly detection, Yogāsana, Bharatnāṭyam, canny edge detection, freeman chain code, state model, Kinect, skeleton data, depth data.*

# LIST OF FIGURES

# LIST OF TABLES

# ABBREVIATIONS

| | |
|---|---|
| HMRA | Human Motion Recognition and Analysis |
| DTW | Dynamic Time Warping |
| HMM | Hidden Markova Model |
| ToF | Time-of-Flight |
| DoF | Degrees-of-Freedom |
| MLD | Moving Light Displays |
| MEI | Motion-Energy Image |
| MHI | Motion History Image |
| MMCRR | Multi-Modality information Collaborative Representation and Recognition |
| BoCPs | Bag of Correlated Poses |
| SVM | Support Vector Machine |
| KHRD | Kinect-based system for Home-based Rehabilitation Dataset |
| DBN | Dynamic Bayesian Networks |
| CHMM | Coupled Hidden Markov Model |
| RGBD | Red Green Blue Depth |
| NBNN | Naïve-Bayes-Nearest-Neighbour |
| CFGs | Context-Free Grammars |
| SCFGs | Stochastic Context-Free Grammars |
| CAVIAR | Context Aware Vision using Image-based Active Recognition |
| ViSOR | Video Surveillance Online Repository |
| K3DA | Kinect 3-D Active |
| BARSDD | *Bharatnāṭyam Adavu* Recognition System from Depth Data |
| BMI | Body Mass Index |
| CED | Canny Edge Detector |
| FCC | Freeman Chain Code |
| DFCCE | Directional Freeman Chain Code of 8 -Directions |

# SANSKRIT WORDS USED IN THESIS

| **Sanskrit Words** | **संस्कृत** |
|---|---|
| *Yogāsana* | योगासन |
| *Samasthiti Tādāsana* | समस्थिती ताडासन |
| *Ūrdhva Hastāsana* | ऊर्ध्व हस्तासन |
| *Vīrabhadrāsana* | वीरभद्रासन |
| *Ūrdhva Baddhanguliyāsana* | ऊर्ध्व बद्धोंगलीयासन |
| *Utkaṭāsana* | उत्कटासन |
| *Vṛukṣāsana* | वृक्षासन |
| *Ardhauttānāsana* | अर्ध उत्तानासन |
| *Yoga* | योग |
| *Yuj* | युज |
| *Āsana* | आसन |
| *Ashtānga Yoga* | अष्टांग |
| *Yama* | यम |
| *Niyama* | नियम |
| *Prāṇāyāma* | प्राणायाम |
| *Pratyāhāra* | प्रत्याहार |
| *Dharanā* | धारणा |
| *Dhyāna* | ध्यान |
| *Samādhi* | समाधी |
| *Bahirang Sādhanā* | बहिरंग साधना |
| *Antarang Sādhanā* | अंतरंग साधना |
| *Antarātmā Sādhanā* | अंतरात्मा साधना |
| *Bharatnāṭyam Adavu* | भरतनाट्यम |
| *Ramāyana* | रामायण |
| *Mahābhārata* | महाभारत |
| *Bhāva* | भाव |
| *Mudra* | मुद्रा |
| *Tatta* | तत्त |
| *Natta* | नट्ट |
| *Sarikal* | सरिकल |
| *Visharu* | विश्रु |
| *Kudittāmeṭṭā* | कुदितमेट्टा |
| *Dāssiattam* | दासीटम |
| *Devdāsi* | देवदासी |
| *Bhāv* | भाव |
| *Tāl* | ताल |
| *Rāg* | राग |

| | |
|---|---|
| *Nṛtta* | नृत्त |
| *Hasta Mudrā* | हस्त मुद्रा |
| *Patāka* | पताक |
| *Kaṭakāmukha* | कटकामुख |
| *Sūcī* | सूची |
| *Udghatita* | उद्घटित |
| *Samachaiva* | समचैव |
| *Āñcita* | अंचित |
| *Mandali* | मंडली |
| *ArdhaMandali* | अर्धमंडली |
| *Sthānaka Mandali* | स्थानकमंडली |
| *Vilambita* | विलंबित |
| *Madhya* | मध्य |
| *Dritta* | द्रत |
| *Tripatāka* | त्रिपताक |

Chapter 1

# INTRODUCTION

# 1. INTRODUCTION

*"Well begun is not only half done, but often fully cooked."*
*-Austin O'malley, Keystones of Thought*

Vision based human motion recognition is a systematic approach to understand and analyze the movement of people in camera captured content. This chapter content brief overview of human motion recognition and analysis (HMRA), inter-disciplinary potential applications, main motivation behind selection of this domain, research problem and objectives are explained. Research contributions and thesis organization are discussed in last sections.

## 1.1. Human Motion Recognition

HMRA comprises of fields such as Biomechanics, Machine Vision, Image Processing, Artificial Intelligence and Pattern Recognition. It is an interdisciplinary challenging field having vast applications with social, commercial, and educational benefits. Human movement can be captured using motion measuring systems with different sensing principles like optical, magnetic or inertial. Use of special kind of suite or markers provide high precision, and avoids human tracking and segmentation. But these systems are expensive, uncomfortable to wear and may limit user movement. This has made it absolute. However, vision based systems become popular amongst researchers due to its portability.

Human motion recognition and analysis has become very important, due to increasing popularity of ambient living and improvements provided by vision based system. Machine can use camera as optical sense organ and recognize the captured contents using intelligent systems.

The general framework of human motion recognition and analysis system is discussed in this section. Each step is discussed in brief to provide overview of domain. Figure 1.1 shows the steps in human motion recognition and analysis. For the understanding and analysis of ongoing motion, a scene needs to be captured with an appropriate capturing system. There is large variety in capturing devices depending on resolution, range and type of output provided by capturing devices. A human needs to be tracked from the scene for further processing. Tracked humans and their motion should be represented efficiently. The represented features are given as an input for motion recognition. The results of human motion recognition highly depend on the selection of an efficient methodology. This selection is decided by parameters like complexity of motion and the extracted features. Figure 1.1 (a) and figure 1.1 (b) are block diagram and pictorial representation of the steps involved in human motion recognition. Representation and recognition methodologies are decided from tracking and initialization of human body in video. Broad approaches for representation are 2-D kinematic or stick figure, 3-D kinematic or shape model and image model. Initialization of human using kinematic method represents human using features like number of joints, its degree of freedom, limb length etc. Whereas, in image model human is represented as image itself and features like shape or region are extracted and stored.

Further, the recognition approach can be decided from representation as well as complexity of the motion. Simple actions uses sequential or space time single layered approaches. Complicated action requires multi layered approaches. These steps are discussed in detail in Literature Survey.

A large variety of applications have different human motion representation and recognition techniques. Human motion analysis is very general term and application decides the number of body parts involved and duration of movement. Human-computer interaction generally involve only hand gestures, whereas, complicated activity or applications like sports, dance may involve all body parts. Depending on complexity, human motion is conceptually categorized into gestures, actions, activity, interactions, and group activities. Complexity in motion is increased from gesture to group activity. Gestures are just movement of single body part example can be raise hand, nod head etc. Action involves multiple body parts in motion but comparatively it is of less duration and less complex e.g. hand waving or clapping. Activity is a single person’s long duration movement involving multiple gestures e.g. single person’s dance, exercise.

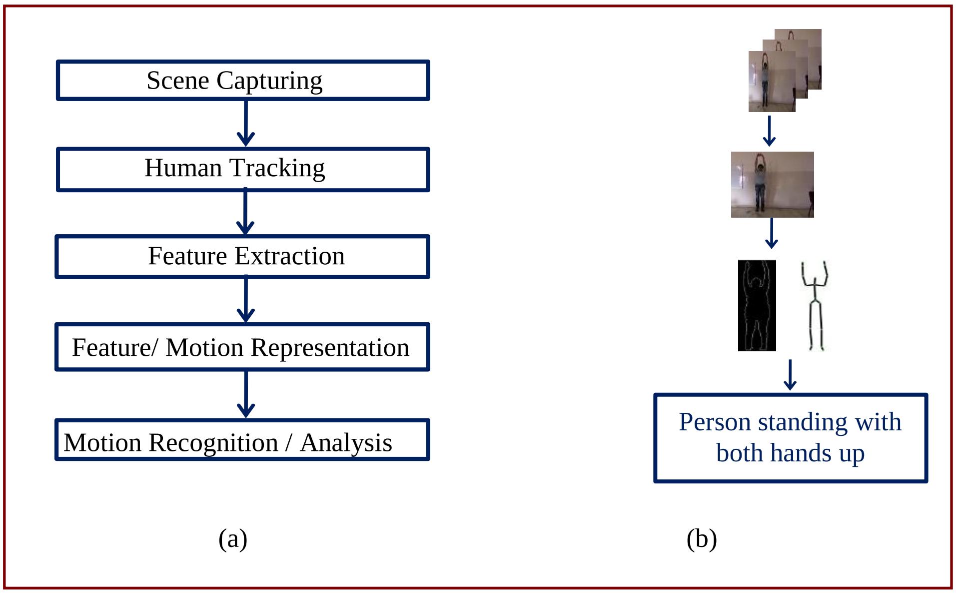



*Figure 1.1. General framework for human motion recognition and its pictorial representation*

Interaction is a movement involving two persons or a person and an object e.g. person drinking water, fighting etc. People shopping at super market can be example of group activity, as it involves multiple persons and objects. Variety of interdisciplinary potential applications and diversity in motion complexity provides different representation and recognition approaches.

Vision based HMRA has made a much progress and many researchers are working in domain at different universities, commercial organizations and research centers. But, practical challenges like in-class variability, shadows, lightning, behaviour understanding, segmentation, modelling, and occlusion handling are yet to be addressed fully. Domain is still far off from becoming an off-the-shelf technology.

## 1.2. Applications

The goal of human motion recognition system is to analyze and interpret automatically ongoing activities from an unknown video. Problem can be visualized as, correct classification of human motion in its activity category. A wide spectrum of application demands human motion recognition. The applications are spread over domains like sports, medical, surveillance, content based video storage and retrieval, man-machine interfaces, video conferencing, art and entertainment, and robotics [1]-[7]. Some of the applications for highlighting the potential impact of human motion recognition are discussed here.

**Smart surveillance:** In today's surveillance systems, video contents are viewed continuously by human operators. With the increasing number of cameras, it is impossible for humans to monitor all the contents 24 X 7. Generally, the contents are viewed after a mishap to analyze the event. So, there is an intense requirement of smart surveillance systems from the security agencies. Smart surveillance systems can analyze an event online and provide appropriate intimation using computer based human motion and behavioural analysis. Smart surveillance is required for access control in special areas like military territory, distant human identification, counting the persons and congestion analysis, detection of abnormal behaviour at shopping malls, railway stations, hospitals, government buildings, commercial premises, and schools [8] [9] [10].

Smart home concept is becoming popular and researchers have designed and implemented application specific smart home systems for old age people, for children etc. [11]

**Behavioural biometrics:** Nowadays, the use of the gait pattern as a biometric has become popular. The main reason is that the recognition of the gait pattern does not require subject cooperation as compared to the other biometrics [12]. Neves J et al. discussed about development of fully automated surveillance system for human identification [10].

**Gesture and posture recognition and analysis:** For a more advanced natural interface with computers and computerized systems, human gesture and posture recognition is an important key. It has promising applications such as gaming, sign language recognition, controlling devices, and others [13] [14].

**Robotics:** Human motion analysis plays an important role in robotics for humanoid robot control, to imitate human motions in a robot in virtual and augmented environments [15].

**Medical:** The medical field uses human motion recognition for the study and analysis of Orthopedic, Neurological and Musculoskeletal disorders. It is also useful in body posture, and fitness. Assistive technologies are becoming popular to improve quality of life of individuals with functional limitations [16]. Intelligent systems to assist elderly people and physically / mentally disabled ones can be designed [17] [18].

**Sports and exercise:** In sports, motion recognition is useful to analyze athletic movements and to design affordable and efficient frameworks for training [19]. An environment for rehabilitation exercise with a feedback system at remote places or in the presence of an expert is designed [20]. Dao designed a monitoring system for the exercises of elderly people [21]. These kinds of systems will definitely be useful for patients and old age people.

**Art and entertainment:** Motion recognition is useful in analyzing, learning, and an emotional understanding of artistic dance movements as in dances such as *Bharatnāṭyam*, *Kathak* and Salsa. Systems to increase the effectiveness of a scene, and the alteration of movements required for quality and the impact of acting can be designed and deployed to use in film industry.

## 1.3. Motivation

Many HMRA systems, algorithms and methodologies with variety of applications are designed and developed by researchers. All the systems are aimed at betterment of the mankind. Design of these systems has challenges such as in-class variability due to human anthropometry, action execution speed, camera view point, shadows, lightning, segmentation, and occlusion handling.

In some applications it is very important to perform actions in specific manner like exercises, traditional dances and acting. Performance of practitioner should be evaluated by expert for correctness. Due to travelling distance, unavailability and time suitability of expert and practitioner, it is not feasible for everyone to perform these actions under expert's supervision. Intelligent systems can be designed to evaluate performance of practitioner and provide feedback in absence of expert. But, this kind of systems could catch attention of very few researchers. The fact motivated to design and implement algorithms for suggestion of amendment in human action. *Yogāsana* dataset is created and taken for testing of these algorithms. *Yoga* is a group of physical, mental, and spiritual practices or disciplines originated in ancient India. *Yogāsana* is one of the limb of *Yoga* deals with physical exercise. In modern age, researchers have experimented and proved that, regular practice of *Yogāsana* gives tremendous benefits in psychological and physiological disorders like blood pressure, diabetes, immune conditions, muscle strength, weight loss, and depression [23] - [25].

*Yogāsana* needs to be performed in recommended manner and it is suggested to be practiced under expert's supervision. But, due to Viewing recorded *Yogāsana* and replicating it may help practitioner, but it does not provide feedback about correctness of their performance. Incorrect practice of *Yogāsana* may lead to other injuries. So, the feedback system for assessment of correctness of *Yogāsana* is very essential. Consideration of *Yogāsana* has social as well as commercial advantage.

## 1.4. Research Statement and Objectives

HMRA domain advances like tracking humans in scene, providing tracked human features in skeleton as well as depth format along with RGB stream provided foundation for advanced algorithms of recognition and suggestion of amendment in human action. The research work on suggestion of amendment in human action is multifaceted and involve factors like in class variations due to anthropometry, execution speed and camera view point.

Systems available for recognition of human action for various applications classify performed action in its correct class and does not comment on incorrect actions. But, for some applications like sports, exercises, traditional dances require analysis of incorrect actions and feedback needs to be given to performer. Worldwide very few researchers worked on this perspective**.** Main stress of this research work is given on designing action execution speed and human anthropometry invariant system.

The prime objective of research work is to propose **"Algorithms for Vision Based Amendment in Human Action with Anthropometric Invariance".**

Objectives of research work are -

1. To carry out extensive survey of literature in human motion recognition.
2. To study and implement existing techniques for representation and recognition of human motion.
3. Data collection and annotation.

4. To design a model for human motion recognition with focus on anthropometric factors using state of the art algorithms.

5. To propose efficient algorithm for human action recognition with anthropometric invariance for unsequestered data pieces.

6. To implement and test algorithm of human motion recognition and provide correction in human motion.

*Yogāsana* dataset is created and tested on designed and developed algorithms. While performing *Yogāsana*, evaluation of movements in terms of kinematic analysis is very important. Availability of vision based systems that provides feedback of amendment in action performed will definitely motivate practitioner for performing exercises at their convenient places and time. System can also keep visual record of exercises performed. This will be helpful for expert to analyze progress of practitioner.

The ultimate goal of system is to provide assistance for person to perform *Yogāsana* at home in absence of expert. System will provide flexibility of time and comfort to perform *Yogāsana*.

## 1.5. Research Contributions

System for suggestion of amendment in human actions is designed and implemented and named as *e-YogaGuru*. System is tested on *Yogāsana* dataset.

This research work contributes the following novelties in the field of human motion recognition -

- Designed specialized state model for *Yogāsana* representation provides advantage of speed invariant recognition of *Yogāsana* (human action).
- Designed and implemented two layer human motion recognition system with execution speed and anthropometry invariance.
- Designed and implemented three layered model for suggestion of amendment in human action and applied it for *Yogāsana*.
- Designed, implemented and tested novel algorithms for vision based recognition of *Yogāsana* and suggestion of amendment in erroneous *Yogāsana*.

- Suggested audio-visual amendment in erroneous *Yogāsana* at two levels
  - Abstract level
  - Detailed amendment
- Comparative study of features provided by Kinect 360 and Kinect 1.0 for *Yogāsana* data.
- *Yogāsana* Replay functionality of performed *Yogāsana* helps practitioner to analyze erroneous or missing postures and expert to monitor progress.
- Created *Yogāsana* dataset and *Bharatnāṭyam Adavu* dataset.
- Designed and implemented *Bharatnāṭyam Adavu* recognition system using depth data for study and analysis of image features.

Any action can be represented using sequence of distinct postures and postures are viewed as states. *Yogāsana* can also be represented using simple state transition diagram. Special kind of pattern is observed in all *Yogāsana* and using this fact simple state model is further modified for recognition of *Yogāsana.* Use of state model provides advantage of speed invariance and flexibility of identification of incorrect posture and suggestion of amendment in terms of body angles at specific posture position with frame number range for erroneous *Yogāsana*. This state model is further modified for suggestion of amendment at incorrect posture positions in erroneous *Yogāsana*.

For recognition of long duration video sequence, single layered approaches are not suitable. Two layered hierarchical approach with execution speed and anthropometry invariance is designed and implemented. Two layered *Yogāsana* recognition model is extended using third layer for suggestion of amendment and system is named as *e-YogaGuru*. After assessment of correctness an amendment is suggested for incorrect *Yogāsana*. Amendment in incorrect *Yogāsana* is suggested at two levels - abstract level and detailed amendment.

**Abstract level amendment** provides details of missing or erroneous posture in required posture sequence in terms of posture name and position.

**Detailed amendment** provides details of practitioner's body angle at erroneous postures along with correct required body angle values and video frame positions.

Algorithms for recognition of *Yogāsana* using joint position feature and joint angle feature are designed, implemented and tested on created *Yogāsana* dataset. An algorithm for assessment of correctness in *Yogāsana* and suggestion of amendment in incorrect *Yogāsana* using angle features is designed, implemented and tested. Replay facility of performed *Yogāsana* helps practitioner to analyze erroneous or missing postures. It also helps expert to monitor progress of practitioner.

Dataset for five standing and two forward bending *Yogāsana* is created in the controlled environment in the form of human body joint positions. Five *Yogāsana* with standing posture are *Samasthiti Tādāsana* (Mountain Pose), *Ūrdhva Hastāsana* (Upward Salute), *Vīrabhadrāsana-II* (Warrior Pose-II), *Vṛukṣāsana* (Tree Pose), and *Ūrdhva Baddhanguliyāsana* (Upward bound hands pose). Forward Bending *Yogāsana* are *Utkatāsana* (Chair pose), and *Ardhauttānāsana.*

Twenty-five practitioners performed three *Yogāsana i.e. Samasthiti Tādāsana, Ūrdhva Hastāsana and Vīrabhadrāsana-II* captured using Kinect 360. Total 750 video sequences (25 practitioner X 3 *Yogāsana* X 10 repetitions) are captured using Kinect 360. Nine practitioners performed six *Yogāsana* (all except *Samasthiti Tādāsana*) captured using Kinect 1.0. Total 540 video sequences (9 practitioner X 6 *Yogāsana* X 10 repetitions) are captured using Kinect 1.0. Total number of video sequences using both sensors are 1290 (i.e.750 + 540) of average length 1800 frames. Out of which in 645 video sequences *Yogāsana* is done correctly and in another 645 video sequences has erroneous *Yogāsana* postures. Total frames considered for recognition are approximately 23, 22,000 frames. Total duration of video sequences is 21 hours 50 minutes. Dataset is created using both Kinect 360 and Kinect 1.0. Kinect 1.0 provides more accurate information.

*Bharatnāṭyam* Adavu recognition system is designed and developed to study and compare humanoid image model approaches with humanoid body model approaches. *Bharatnāṭyam* dataset is created in the form of depth stream using Kinect sensor and tested for five *Adavu*. Main objective behind this work is system should recognize meaning of dance and display it for audience. We have represented *Adavu* using extended state model. Five basic *Adavu Tatta*, *Natta*, *Sarikal*, *Visharu*, and *Kudittametta* are considered for recognition.

*e-YogaGuru* system designed, developed and implemented for suggestion of amendment in *Yogāsana*, helps practitioner to assess the improvement and progress in performing *Yogāsana*. It reduces economic burden and travel time. Expert can change knowledge-base for specialized practitioners. In future designed algorithms can also be applied for other applications like sports, different exercises and traditional dances by changing knowledge-base and state transition model.

## 1.5. Organization of the Thesis

*e-YogaGuru* system is designed, implemented and tested for created *Yogāsana* dataset. All the defined research objectives are achieved. Research work is described in this thesis in five chapters and contents of each chapter are discussed in brief in subsequent section.

**Chapter 2 Literature Survey** gives bird's eye view of domain to readers. Highly relevant state-of-the-art literature is discussed in chapter. Important components of human motion recognition are discussed in detail. Chapter is concluded with discussion on identified challenges in domain and need of more research.

**Chapter 3 Model Design and Algorithms for Vision Based Amendment in Human Action with Anthropometric Invariance,** first discuss about *Yogāsana* dataset creation along with capturing environment and feature selection. Further, two layer and three layer hierarchical approaches along with design of simple state models and modified state model for recognition of *Yogāsana* and suggestion of amendment are elaborated. Designed algorithms for *Yogāsana* recognition and suggestion of amendment and recognition of *Bharatnāṭyam Adavu* recognition are explained.

**Chapter 4 Results and Analysis,** provides details about experimentations done and obtained results, tested on *Yogāsana* and *Bharatnāṭyam* dataset.

**Chapter 5 Conclusion and Future Scope** explains the conclusion and future scope of suggestion of amendment in human action.

# Chapter 2

# LITERATURE SURVEY

# 2. LITERATURE SURVEY

*``If I have seen further than others, it is because I've stood on the shoulders of giants."*

*- Sir Isaac Newton*

A significant development in human motion recognition and analysis has been seen in the last two decades, and a plethora of literature in the form of journals, transaction papers, patents, reviews, and surveys is available. To provide an overall idea of the domain, the main stages in HMRA and highly relevant literature are described in subsequent sections. First, all the major components of system are explained using mind map. Then scene capture, human tracking, human motion representation and human motion recognition along with selected datasets are discussed. Chapter is concluded with challenges in domain and need of more research.

## 2.1. General Overview of Human Motion Recognition

To classify the previous work in the domain, the researchers have used criteria like type of models (e.g. stick figure-based, volumetric, statistical) and the dimensionality of the tracking space (2-D Vs 3-D) [4] [5]. Some reviews classify the literature using complexity of the action to be identified (e.g. gesture, action, interaction, group activity) [3]. Some survey uses sensor modality (e.g., visible light, infra-red, range), sensor multiplicity (monocular Vs stereo), various applications, number of persons, number of tracked limbs, and assumptions (rigid, non-rigid, elastic) for the classification of the available literature [1] [2] [9] [26].

Aggarwal is active in human motion recognition domain since the 1970s, and with a significant amount of work, he has provided time to time reviews of the domain [1] - [3]. In his recent review paper with M. S. Rhoo, human motion recognition approaches are classified in to two groups: Single layered approaches and Hierarchical approaches [3].

Cedras and Shah have described motion based recognition in two steps, the motion information is extraction at first step, and the second step is the matching of an unknown input with the constructed model [28]. Gavrila has classified the literature into three categories: (a) 2-D approaches without explicit shape models, (b) 2-D approaches with explicit shape models (usually stick figures, wrapped around with ribbons or “blobs”) and (c) 3-D approaches (surface-based or volumetric) [4]. T. B. Moeslund has given a compressive survey of publications in computer vision-based human motion analysis in the form of two survey papers [26] [28]. Taxonomy used in both the survey papers is initialization, tracking, pose estimation, and recognition. The author has classified human motion related applications into surveillance applications (e.g. people counting, congestion analysis), control applications (e.g. Human Computer Interfaces, advanced gaming) and analysis applications (e.g. automatic analysis of orthopedic patients, and content based retrieval). Wang, Hu, and Tan proposed a three stage framework for human motion analysis, i.e. human detection (motion segmentation, object classification), human tracking (Model based, Region based, Active contour, Feature based) and human behavior understanding (Dynamic Time Warping (DTW), Hidden Markova Model (HMM), Neural Network, and Semantic description) [29]. Neves J. et al. discussed advances in both human motion analysis and biometric recognition for the development of fully automated surveillance system [10]. Paper describes human detection, human tracking, and human motion recognition techniques with special efforts to address surveillance scenarios.

Figure 2.1 shows a general overview of the components of a human motion recognition and analysis system. They include a scene capturing system, human tracking, motion representation methods, motion recognition methods, applications, and datasets. The details of each component are discussed in subsequent sections. Applications have been explained in chapter 1. Introduction. The sample papers for discussion are selected such that they provide a small tour of the domain development. Each component of system is discussed in detail in subsequent sections.

## 2.2. Scene Capture

The selection of a capturing system depends on the application. Surveillance system applications require more area coverage, whereas a device interface requires a more

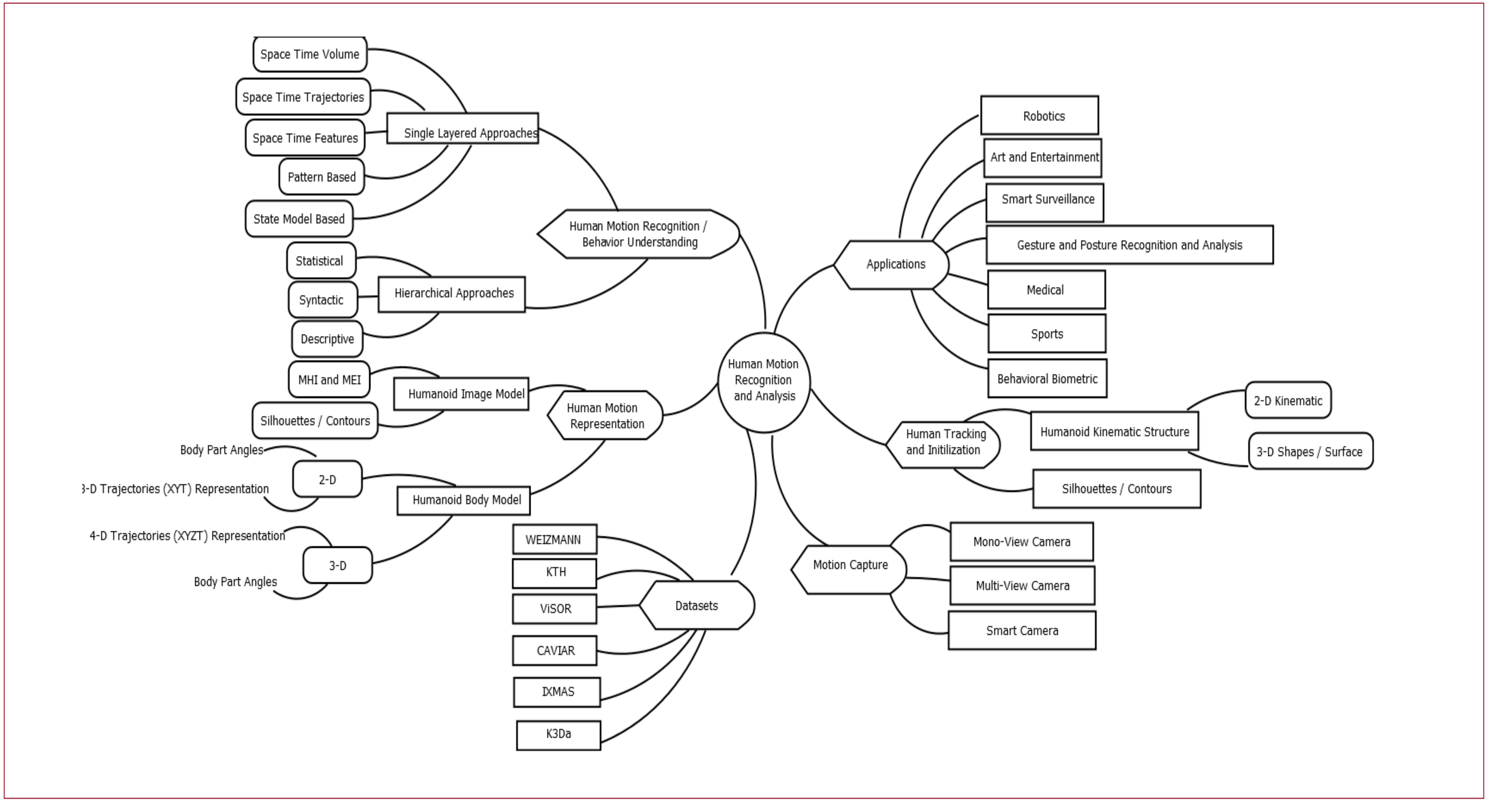



*Figure 2.1: Components of human motion recognition*

detailed analysis of motion. Nowadays, high resolution cameras are available for surveillance applications. Analysis and controlling applications are either captured by single camera or multiple camera systems. The contents that are captured with a single camera are easy to process, but they may miss the detailed human features. However, the contents that are captured with multiple cameras provide detailed features, but make the systems complicated for processing and analysis. Important initial parameters of human motion recognition and analysis system are deciding environmental, subject constraints and camera calibration. Subject constraints can be clothing, known or new subject, subject trained or not trained for performing actions etc. Environmental constraints include illumination details, constant or variable background etc. In order to process scene contents camera details like resolution, frequency of frames, single camera Vs multiple cameras, camera view point etc. need to be considered. With the advancement in the domain, many sophisticated capture systems like *Microsoft Kinect*, and *ASUS Xtion Pro Live* are available. Such systems simplify the initial stages of capturing and human tracking. They provide output in the form of human skeleton joint location details, depth data of objects etc. Researchers can apply representation methodologies and recognition algorithms on obtained data for motion recognition and analysis.

## 2.3. Human Tracking

Human tracking is the identification and initialization of humans in images or videos. It estimates the location and scale of people in images. Human tracking is challenging, but it is the most important task for the recognition of human motion. E. Marinoiu, D. Papava and C. Sminchisescu have constructed an experimental apparatus that tracks eye movement recording of human subject on 2D and 3D ground truth data [30]. They extensively analyses perception by human and poses recognized by automatic system by providing the same visual stimuli.

They have concluded that when perceptual visual learning approach is integrated with visual sensing system, it produces more stable and meaningful results.

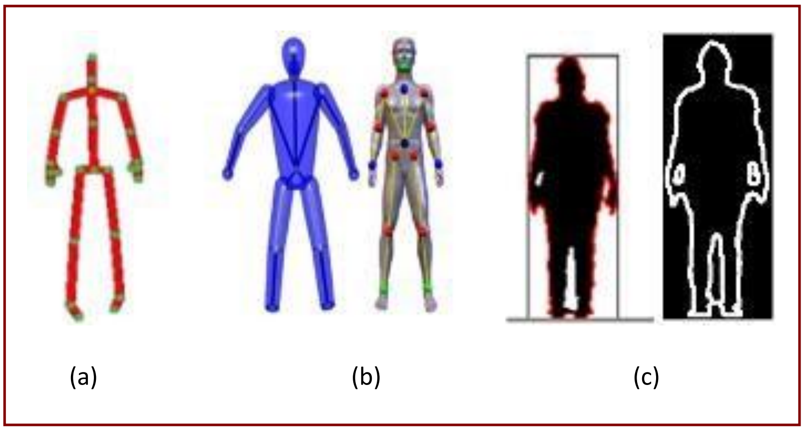



*Figure 2.2: Initialization Models: (a) Humanoid Kinematic Structure, (b) Humanoid Kinematic shape structure (c) Silhouettes and Contour*

Nowadays, smart cameras are available in the market, which track four to six humans in an indoor scene. Kinect is most popular amongst them and two versions of it are available in market. Table 2.1 shows comparison of features of both the cameras. Kinect tracks humans in a scene and provides its depth data and skeleton stream as an output for further processing [31]. The literature has already reported the correctness of depth and the skeleton data of Kinect. Khoshelham K., and Elberink have analysed correctness of the depth data. Quality analysis of depth data shows that random error in depth increases with increase in distance from few mm up to 4 cm for 5 m range. Obdrzalek et al. analyzed accuracy and correctness of Kinect pose estimation [32] [33]. B. Galna et al. analysed accuracy of Kinect sensor for measuring movements in people with Parkinson's diseases [34]. They have compared results of Kinect with Vicon system with 10 MX3+ infrared camera. Authors concluded from their work that Kinect

*Table 2.1: Comparison of Kinect Xbox 360 and Kinect 1.0*

| Feature | Kinect Xbox 360 | Kinect 1.0 |
|---|---|---|
| **Colour Camera** | 640x480 @30fps | 1920x1080 @30fps |
| **Depth Camera** | 320x240 | 512x424 |
| **Max Depth Distance** | ~ 4.5M | ~ 4.5M |
| **Min Depth Distance** | 40 cm in near mode | 50 cm |
| **Horizontal Field of View** | 57 degrees | 70 degrees |
| **Vertical Field of View** | 43 degrees | 60 degrees |
| **Tilt Motor** | Yes | No |
| **Skeleton Joints Defined** | 20 joints | 25 joints |
| **Full Skeleton Tract** | 2 | 6 |
| **USB Standard** | 2.0 | 3.0 |

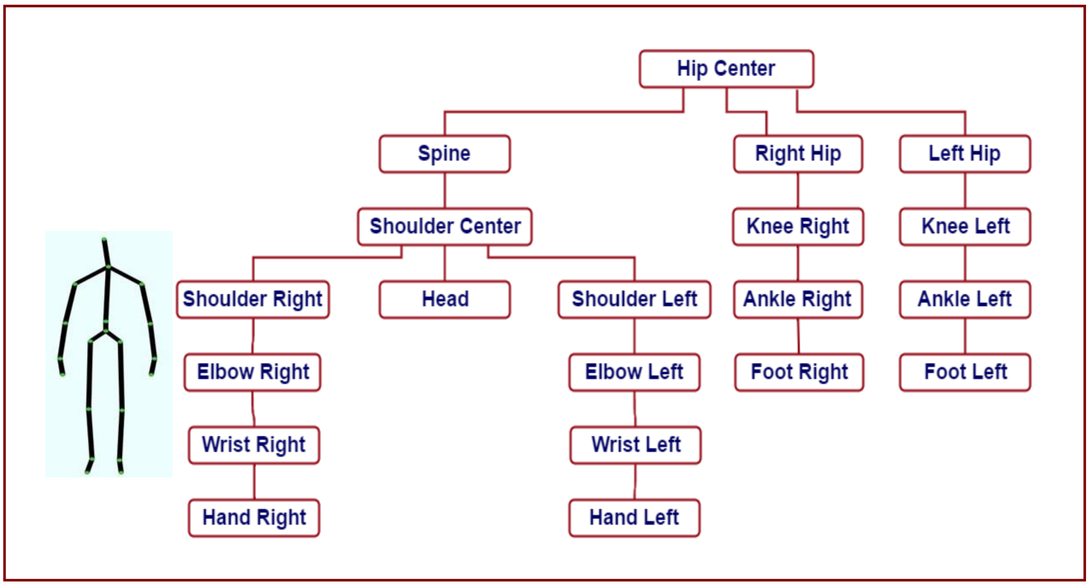



*Figure 2.3: (a) Skeleton structure and joint points obtained using Kinect 360*

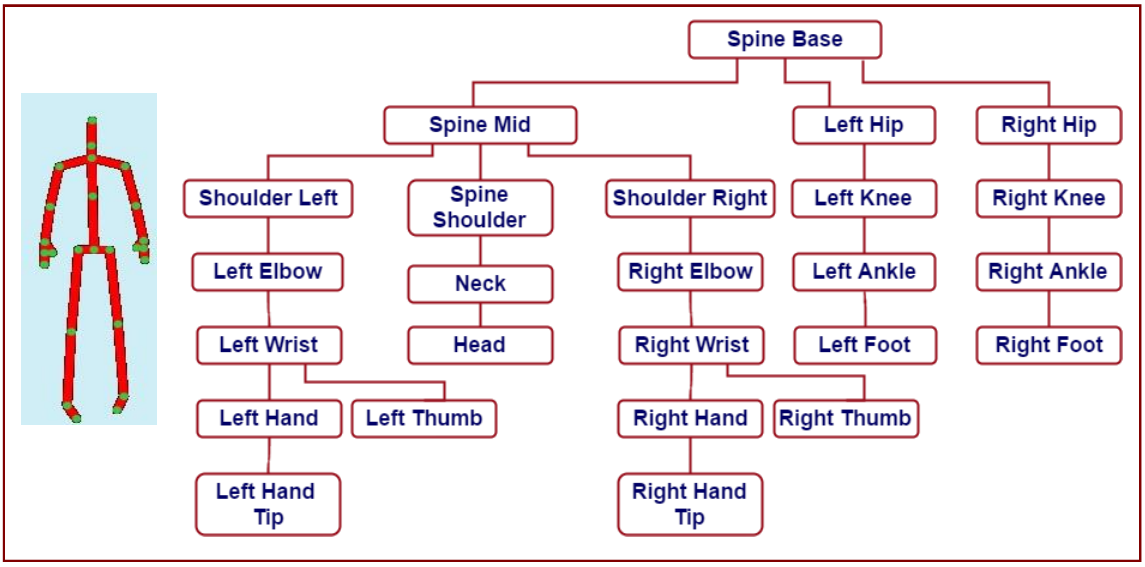



*Figure 2.3 (b): Skeleton structure and joint points obtained using Kinect 1.0*

can accurately measure timing and spatial movement characteristics for gross movements, but gives poor performance for fine movement. J. Smisek's experimental results of 3-D measurement capability for three different cameras; Kinect, a stereo, and a Time-of-Flight (ToF) camera reveal that Kinect is superior in accuracy to the ToF camera and close to a medium-resolution stereo camera [35]. T. Stoyanov compared performance of Kinect and 2 ToF 3-D camera with ground truth data produced by laser range sensor, and has shown that Kinect has better performance than TOF and very close to laser range for short distance [36].

Many researchers have provided human action recognition algorithms using the depth data and skeleton streams obtained from the Kinect sensors [37] [38]. The use of Kinect provides assistance for the low level vision problem i.e. scene capture and human tracking. These kinds of robust solutions in the form of products help the researchers to concentrate on high level vision problems. Details of both Kinect version is discussed in subsequent section as developed system uses Kinect skeleton data as input for designed algorithms. Kinect 360 provides 20 joint positions, whereas Kinect 2.0 gives 25 joint positions. Skeleton stream details obtained with Kinect 360 are shown in figure 2.3 (a) and skeleton stream details obtained with Kinect 1.0 are shown in figure 2.3 (b).

## 2.4. Motion Representation Methods

Human action recognition and representation methods are interdependent. The representation methods are broadly categorized into the humanoid body model and the humanoid image model. The humanoid body model uses kinematic structure, and the image model uses humanoid shapes or contour.

### 2.4.1. Humanoid Body Model

The humanoid Body Model uses structural representation and represents a person using his joint positions as a set of 2-D (X, Y) or 3-D (X, Y, Z) points in space. The modelling of a stick figure uses human body parts as an estimation methodology. It helps to extract the joint positions of a person from image frame. Kinect 360 provides (X, Y, Z) points of the human skeleton data for 20 joints, whereas Kinect 1.0 provides 25 human joints points as shown in figure 2.4(d) [39]. A different number of joints and their Degrees-of-Freedom (DoF) are considered for representation of the human pose. Generally, 20 DoF is used but, for a more detailed analysis like consideration of twisting movement, 34 DoF to 50 DoF models are used. Changes in the person's joint positions while performing an action are stored as space-time trajectories. 3-D XYT or 4-D XYZT space time trajectories are used for the representation of these actions [40]. Johansson is a pioneer in finding components that provide information to humans for understanding human motion [41].

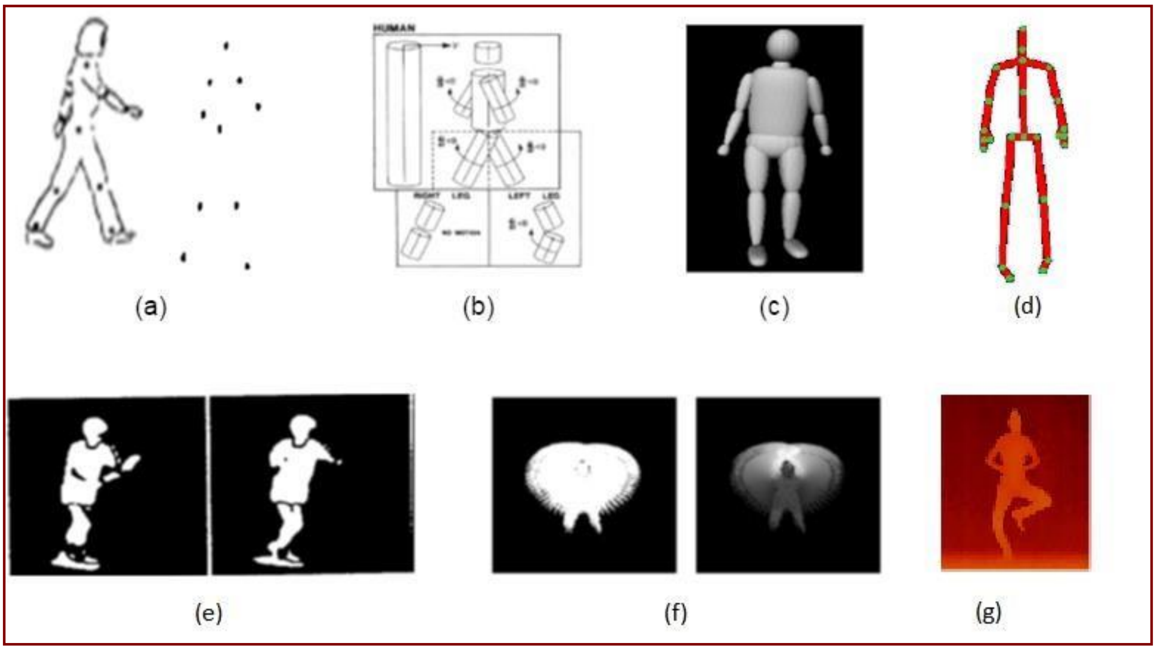



*Figure 2.4: (a)-(d) Humanoid Kinematic Structure [39][41][42], (e)-(g) Humanoid Shape Model [43][44]*

He has experimented it by attaching moving light displays (MLD) to the human body parts. He has recommended that the tracking of joint position is sufficient for humans to distinguish between actions as shown in figure 2.4 (a). However, he has not worked on camera captured contents. Use of human joint points is the best possible feature that provides recognition of motion. Shotton et al. have extracted 3-D position of body joints from single image [42]. They have estimated human body parts invariant to pose, body shape, clothing -etc. using their large highly varied training dataset.

In several video-based human motion recognition systems the humanoid body model is used to estimate a person's shape. Generally, the model is constructed using shape primitives or surface primitives. Cylinders, cones, ellipsoids, and super quadric are general shape primitives which are used for humanoid shape construction. Polygonal mesh, subdivision surface etc. are surface primitives used for humanoid shape reconstruction. Figure 2.4(b) shows a humanoid model constructed from cylinders as suggested by Marr and Nishihara [43]. Sminchisescu and Triggs used a super quadric ellipsoids to represent flesh on skeleton of articulated as shown in figure 2.4(c) [44]. A major advantage of the humanoid body model representation is that it is camera view invariant.

### 2.4.2. Human Image Model

Humanoid image based representation approaches are also known as holistic representation. In these approaches, an action is represented as an image and it does not require detection of an individual body part. Silhouettes or contours of humans performing the action are used for representation. Yamato et al. have represented a person using silhouettes as shown in figure 2.4(e) [45]. They have used ratio of the number of black pixels to the number of white pixels in a mesh as feature for representation.

Bobick and Davis represented human action using a 2-D Motion-Energy Image (MEI) represented in binary format and a Motion History Image (MHI) [46] [47]. MHI is constructed by projecting a sequence of foreground 2-D images on 3-D space-time volume as shown in figure 2.4(f). MHI represents image sequence using decreasing weightage to the sequence of images with less weight to older frames and more weight given to new frames. MEI gives equal weight to all the images in the sequence. Z. Gao has used fusion of depth-MHI and RGB-MHI Multi-modality information collaborative representation and recognition (MMCRR) model [37]. Depth data used as input is obtained using Kinect. Depth data of human posture obtained using Kinect is shown in figure 2.4(g). They have experimented their approach on MSR Action3D dataset DHA – contain 17 actions performed by 21 people. It shows an average recognition rate of 95.2%.

A major disadvantage of image based approaches is that they are very sensitive to action change, as well as view and size variability.

## 2.5. Motion Recognition Methods

Human motion can be determined by taking difference between two pixel values in consecutive frames. Kinematic approach represents motion trajectory by 3-D trajectory points $(X, Y, T)$ or 4-D trajectory points $(X, Y, Z, T)$. Each point corresponds to respective joint value in frame for human posture. Image or shape approach represents motion using optical flow or using MHI or MEI. Human motion can be either directly recognized from image sequences, or it can be done in a multiple layer process. Generally, for simple actions, motion is recognized directly from image sequences and they can be viewed as single layer approaches. However, complicated activities can be recognized by using multiple layer recognition methods. Gestures and actions can be recognized using single layer approaches. Complicated motion i.e. activities, interactions, and group activities are generally recognized using multiple layers, by decomposing it into simple actions or gestures. Recognized simple actions or gestures at lower levels are used for the recognition of complicated motions at higher levels. Activities like fighting can be viewed as a sequence of punching and kicking. In the lower layer, the atomic actions of punching and kicking are identified, and then at a higher level an activity is recognized by the sequence of atomic actions.

The recognition methods for simple actions are categorized into space time volume, space time trajectories, space time local features, pattern-based approaches, and state space based approaches. However, for complicated activities and interactions, multi-layer recognition methods are applied. Multi-layer recognition approaches are statistical approaches, syntactic approaches, and descriptive approaches. These approaches are discussed in detail in the subsequent sections.

### 2.5.1. Trajectory Based Recognition

Trajectory based approaches represent human information using the Humanoid Kinematic Structure. The Humanoid Kinematic approach extracts different features like joint angles, limb lengths, and DoF from each key-frame. Sheikh, Y., Sheikh, M., and Shah, M have represented an action using 4-D XYZT space trajectories for 13 joint [48]. X,YT trajectories of an action are obtained using affine projection. That in turn helped to find the view-invariant similarity between two sets of trajectories as shown figure 2.5 (a). Champbell and Bobick have represented the human action as curves. Joint positions are tracked using 3-D human body models [49]. For each frame, a body is defined using 3-D estimated models. Static pose in a frame is represented using set of points and action is represented by curves. Each curve is modelled as a cubic polynomial form. The approach is tested on independent pieces of ballet dance. The systems gave speed invariant recognition with approximately zero false acceptance.

### 2.5.2. Space Time Volume

An activity execution in a video can be represented using 3-D space-time volume assembled by concatenating 2-D images with respect to time. There are different

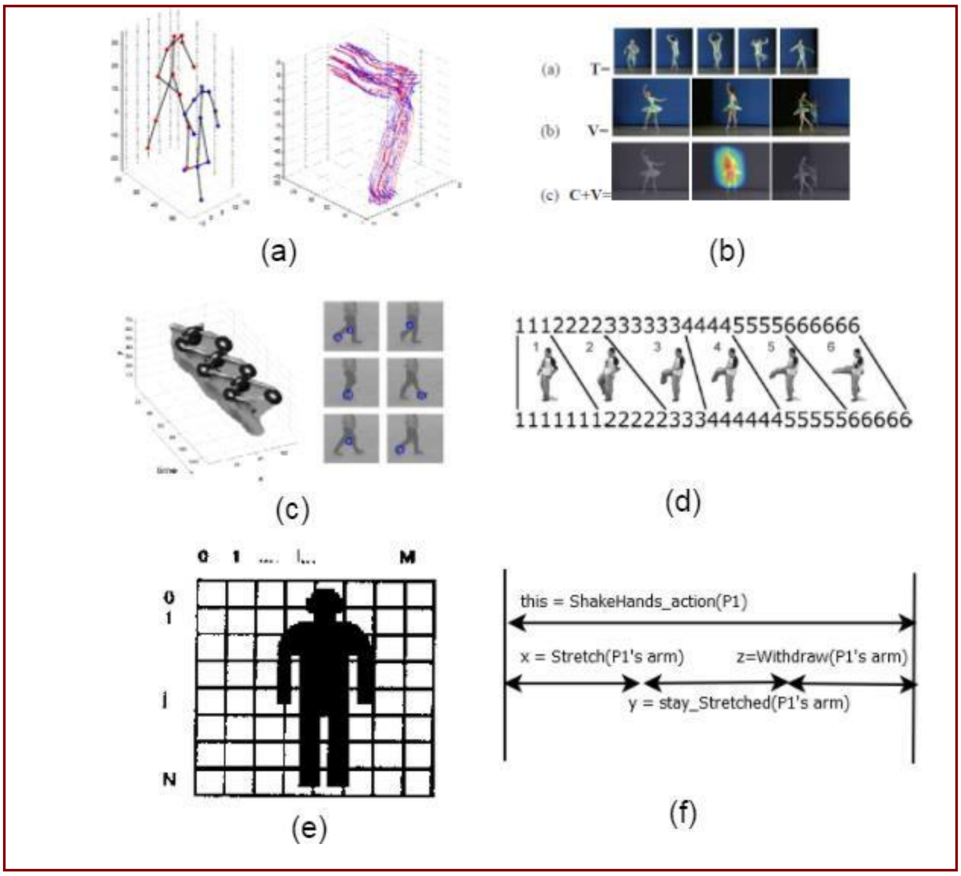



*Figure 2.5: Human motion recognition methods using (a) 4-D XYTZ approach [46], (b) 3-D space time volume [49] ,(c) space time local feature [50] , (d) dynamic time wrapping [3], (e) HMM using number of pixels in each mesh  feature [43], (f) statistical recognition techniques [57]*

methods used by the researchers to store the video in the space-time-volume fashion. Figure 2.5 (b) shows space time volume considered for recognition of action. Bobick and Davis have stacked extracted the human silhouettes to track the shape changes as shown in figure 2.4 (f) [47]. They have used template matching approach for recognition of human activity in kids' room application. Bobick has further suggested that MEI and MHI are not suitable for complex activities due to overwriting of the motion history [46]. They give unreliable output for complex activities.

Wu and Shao have fused the temporally local descriptors called the bag of correlated poses (BoCPs) [50]. Gait energy information (GEI) and inversed recording (INV) are added to the MHI. Loss of information in MHI is compensated by GEI and INV. BoCP is created using k-means algorithm. Here, the feature vectors of all the training samples are clustered. The centre of each cluster is defined as a code word and the number of visual code words is the number of the clusters in visual vocabulary. The authors claimed that their approach produces improved results for IXMAS dataset as compared to others.

Shechtman and Irani used a set of 3-D space time volume segments corresponding to a moving human as shown in figure 2.5 (b) [51]. Hierarchical similarity measurement on the space-time volume correlation is used for recognition. The approach is tested on different video databases with the video as the query, like different walking patterns on a beach, dives in a pool, and ballet footage.

### 2.5.3. Space Time Local Features

Local features from each frame are extracted in space time local feature method. They are concatenated for all time to describe the overall motion of human activities. The local features are also referred to as local temporal feature-based approaches. These approaches have shown success in case of change in illumination and presence of noise. Multiple activities are recognized without constructing a human model and subtracting the background.

Laptev and Lindeberg have applied the space time feature based approach for walking in an outdoor scene [52]. Extracted spatio-temporal interest points are represented on XYT as shown in figure 2.5(c). The Support Vector Machine (SVM) is used for the classification of motion. More complex activities cannot be modelled using Space-time feature-based approaches.

### 2.5.4. Pattern Based

Pattern-based approaches for human activity recognition use whole activity sequence for matching. Sequences of the feature vectors extracted from the query video are compared with templates or sample action executions. Template matching requires major attention towards variability issues. Algorithms like DTW can be used for speed invariant action detection. Veeraraghavan, A., Chellappa, R., and Roy-Chowdhury, A. K. have applied DTW for human motion recognition in human actions with different speeds for activities like throwing an object, picking an object, pushing, and waving [53]. Observed recognition accuracy is very high for all these activities. Marr and Nishihara used DTW for recognition of walking pattern. It has been tested on real as well as synthetic data of a pedestrian [43]. 3-D motion trajectories of body parts are used as feature for the recognition of motion. Chuan-Jun Su has developed Kinect-based system for home-based rehabilitation (KHRD) for patients [54]. System uses DTW to compare (X, Y, Z) values of two exercise sequences. It provides output in the form of Excellent, Good, and Bad depending on execution speed and trajectory matching. Their system gave 80.01% results. System uses exercise performed in presence of expert as base for evaluating the exercises performed at home. Their system does not provide generalized solution. Generally, for patients there is always day by day improvement in performing the exercise and after practice one comes to perfectness. But, discussed KHRD consider same persons exercise as reference.

### 2.5.5. State Model Based Approaches

State model based approaches represent a human activity using a set of states. The model is statistically trained for corresponding activity class feature vectors. Probabilistic models like Hidden Markov Models (HMMs) and dynamic Bayesian networks (DBNs) are used for recognition.

Yamato et al. have applied HMM for human motion recognition for the first time [45]. The foreground is converted into meshes and the number of pixels in each mesh is considered as a feature. This approach is applied for the recognition of 6 tennis strokes and 3 persons have performed each tennis action 10 times. Five sequences are used to train HMM and five are used to test the recognition performance. Major disadvantage of these approaches is that new model needs to be designed for each considered action.

Lin and Kulic have segmented on line data and also identified human action [18]. For segmentation the starting and ending point of each action is correctly identified from data. Their approach velocity peaks and zero velocity crossings segmentation of motion. Only the significant DoFs are selected for a given template. Feature-guided Left-right HMM is used in the second phase with joint angles as observation data and hidden states as key poses. The approach is tested for real time interactive feedback in the rehabilitation application for segmentation along with recognition. They have considered rehabilitation movements of 20 healthy persons and 4 rehabilitation patients for test data. Segmentation accuracy achieved is 87% for user specific templates and 79-83% for user-independent templates.

For a complicated activity or interaction, state space based approaches are modelled using multiple layers generally two layers. Simple atomic actions are recognized at the bottom layer from the sequences of the feature vectors. The higher level models treat this sequence of atomic actions as observations, and the maximum likelihood estimation or a maximum posterior probability classifier is constructed to classify the activity. Oliver et al. have given a real-time visual surveillance system for modelling and recognizing human behaviors [55]. Interactions are modelled using both the Coupled Hidden Markov Model (CHMM) and HMM.  The complexity of the various interactions decides number of states per chain and number of states in HMM. Here, CHMM shows better results than HMM.

Al Mansure et al. have applied inverse dynamics to obtain dynamic features for action recognition [56]. These human body features are obtained by using physics-based representation. The dynamic features considered are: the torques obtained from knee and hip joints and gravity, ground reaction forces, and the pose of the remaining body parts. The authors claimed that these features more discriminatively represent human action than the kinematic features. This approach gives good classification results when applied using the HMM classification framework.

M. Yang et al. recognized three actions sit down, stand up and stoop using finite state machine [57]. They have used body angle features obtained from 3-D skeleton data of Kinect. Their approach provided 100% results, but dataset considered is very simple and small. S. Nowozin 2-D kinematic Joint angle velocity and ZVC's, joint angles, and joint angle velocities [58]. Action points representation Firing HMM Random Forest Direct

Classification  Weizmann repetitive actions (93 sequences of 9 actors performing 10 actions) Kinect actions Dataset (54 sequences from 17 actors performing 10 actions) FHMM gave 98% multiclass accuracy for Weizman dataset. For online recognition random forest is superior over F-HMM.

X. Yang and Y. Tian have used eigenJoints obtained from 3-D skeleton joint points obtained from depth data of RGBD cameras [38]. Their experimentation is tested on MSR 3-D dataset, Cornell Human Activity dataset and UCF dataset using Naïve-Bayes-Nearest-

Neighbor classifier. Main contribution of their approach is that it recognizes action within first 30-40% frames as compared to other approaches Bag-of-3D-points and HOJ3D. However performance of their system is outperformed Bag-of-3D-points and almost same as HOJ3D. Naïve-Bayes-Nearest-Neighbor (NBNN) With PCA for eigenJoints Tested on Cornell Human Activity UCF Kinect Dataset for same training and testing data showed that more than 95% in cross subject test half are used for training and remaining for testing.

J. Sung et al. have detected unstructured human activity using supervised learning [59]. They have trained two-layered maximum-entropy Markov model to capture properties of human action. Data is collected on two male and two female for twelve activities. “Have seen” and “new person” actions have shown precision / recall measures as 84.7% / 83.2% and 67.9% / 55.5% respectively. Change in clothing and background cause more confusion to system and can cause confusion and provides precision/ recall value for “have seen” as 33.1%/23.5%.

### 2.5.6. Hierarchical Syntactic Approaches

Syntactic approaches use symbols to represent human activity. Each atomic action is represented as symbol and string of symbols represents the action sequence. Researchers generally use Context-Free Grammars (CFGs) and Stochastic Context-Free Grammars (SCFGs) parsing techniques for the recognition of human activity. Recognition process is designed using more than two layers. Atomic actions are recognized at the bottom layer and are represented as a symbol. At the upper layer, human activities are represented using a set of production rules and are recognized by parsing techniques. These approaches are suitable for a sequential activity, and not for a concurrent activity, but face difficulty when an unknown object appears in the scene.

Ivanov and Bobick have used the SCFG parsing mechanism for the recognition of multiple interacting objects in surveillance and gesture recognition [60]. HMM is used at low level for detection of atomic feature. The outputs of these detectors provide the input stream for a SCFG parsing mechanism. The authors have handled uncertainty in the input symbol stream by extending stochastic context-free parsing and also handled consistent multi-object interactions.

### 2.5.7. Descriptive Approaches

Descriptive approaches represent activities in terms of sub-events / actions with some logical, spatial or temporal relationship. A multi layered design is a must for descriptive approaches. A major advantage of these approaches is that they can handle a concurrent structure.

Ryoo and Aggarwal have given a general framework, for recognition of and human-human interactions and complex human action [61]. The authors used statistical recognition techniques from computer vision and knowledge representation concepts from traditional and artificial intelligence. Figure 2.5(f) shows an example of hand shaking. Based on the recognition of gestures at the low level, the high-level of the system hierarchically, recognizes the composite actions and interactions occurring in a sequence of image frames. A major contribution of the authors is the system recognizing human activities including ‘fighting’ and ‘greeting’, which are high-level activities. They have a very high recognition rate.

A brief overview of all the components of human motion recognition is discussed in above section.  Research has been tested designed algorithms on *Yogāsana* dataset. Main objective is to design and implement algorithms for suggestion of amendment in human action with anthropometric invariance. Subsequent section covers almost all relevant papers with *Yogāsana* or other exercises as dataset along with advantages and disadvantages.

Li Yao et al. have given Kinect-based rehabilitation exercise system with therapist involved approach [62]. Their system consists of three modules (i) Exercise prototyping tool, (ii) Action Evaluation module, and (iii) Data statistic and analysis module. They used cross correlation ($r_{xy}$) for recognition. System is not self-sufficient to give feedback, however it requires regular attention of therapist to provide feedback to user.

Chen H.T et al. have explained a distance computer vision assisted *Yoga* learning system [63]. They have compared *Yoga* practitioner’s posture silhouettes with previously stored correct posture silhouettes. However, their match score with correct expected does not comment on correctness of postures. Authors used star skeleton and does not consider detailed features for in depth analysis of *Āsana* postures. Angle between thigh and leg at knee position is important for considered *Āsana Vīrabhadrāsana - II*. However, it has not been considered in their proposed approach. Also, their system considers only main *Yoga* posture and not the complete sequence. Zimmerman in his US patent has suggested system and methods for motion analysis and feedback [64]. He has used sensors placed at vital energy points and tactile feedback devices provide tactile feedback to indicate errant motions. However, our approach provides in depth analysis of required correction in practitioner's posture. Also consider complete *Yogāsana* sequence for recognition and suggestion of amendment.

## 2.6. Human Motion Datasets

A standardized dataset is an important need of each domain for the comparison and assessment of an algorithmic performance. With advances in the domain, a variety of datasets are available, some contain an indoor scene, and some are with an outdoor scene. They also differ with motion complexities, number of camera views, moving camera, or a still camera. A detailed analysis of a dataset is not the main focus of the paper. This section is added for the completeness of the paper. Chaquet et al. provided the detailed analysis of almost 68 different datasets [65]. They have divided human motion datasets into heterogeneous actions, specific actions and other categories. Here, we have considered only, the mostly referred top few datasets with a variation in the actions for discussion.

WEIZMAN (Event and Action), KTH, CAVIAR, ViSOR, IXMAS and K3DA are considered for discussion.

### 2.6.1. Weizman Datasets

The Weizmann Institute of Science provides two datasets- The Weizmman Event-Based Analysis dataset and Weizmann actions as space-time shapes. The Weizmman event-based analysis dataset was created in 2001. This dataset is used for studying algorithms and segmentation methods.

The Weizmann actions as space-time shapes dataset was created in 2005, to design new algorithms for improving the human motion recognition results. This dataset contains a static background and the moving person's foreground silhouettes. The actions considered and the capturing details are given in Table 2.2.

### 2.6.2. KTH Dataset

The KTH Royal Institute of Technology created this dataset in 2004. There are total 25 X 6 X 4 = 600 video files created using a 25 subjects performing 6 actions, and captured in 4 different scenarios. The sequences were taken at 25 frame rate using static camera and same background. The details of actions considered, and the capturing environment are shown in Table 2.2.

### 2.6.3. CAVIAR Dataset

Context Aware Vision using Image-based Active Recognition (CAVIAR) was the first dataset recorded in more complex environments. The CAVIAR dataset includes 9 activities. The videos are recorded at two different places. The first at the entrance lobby of the INRIA Labs at Grenoble France and the second part is recorded at a shopping centre in Lisbon. The CAVIAR project has produced huge publications that focus on variety of applications, such as target detectors, activity recognition, monitoring of human activity, segmentation, and tracking of movement, or multi-agent activity recognition.

### 2.6.4. ViSOR Dataset

ViSOR (Video Surveillance online repository) dataset is provided by the Image Lab of the University of Modena and Reggio Emilia. It is more realistic and contains videos captured with surveillance cameras in indoor and outdoor scenes. The researchers have used these datasets for the study of human behavior analysis, detection of smoke, human tracking, event analysis, counting people, pedestrian crossing, and identification of human, also human action recognition.

### 2.6.5. IXMAS Dataset

IXMAS is created for view-invariant human action recognition. Ground truth, and silhouettes are provided in a BMP format (390 X 291) and reconstructed volumes (64 X 64 X 64) are provided in the MATLAB format. This dataset contains 13 day-to-day

*Table 2.2: Human motion dataset*

| **Dataset Name, Year** | **Type of Actions Considered** | **Capturing Environment Details** |
|---|---|---|
| WEIZMANN Action 2005 | Walking, waving, running, Running in place. | Mono-View, has variability in subject and their clothing. |
| WEIZMANN Event 2001 | walking, jumping, galloping sideways, bending, waving with single and both the hands waving, jumping on place, jumping jack, and skipping, running,. | The view-point is static.<br>Unrealistic action analysis. |
| KTH 2004 | Walking, jogging, running, boxing, hand waving, and hand clapping. | Mono-View, Non-realistic, Captured in controlled environment. Added complexity by changing lighting conditions and clothing of subjects, unrealistic action analysis. |
| CAVIAR -Context Aware Vision using Image-based Active Recognition 2007 | Meeting, Walking, entering, in shop, browsing, slump, left object alone, fighting, window shop, and exiting from shop. | Captured at the entrance lobby of the INRIA Labs at Grenoble, France and a shopping centre in Lisbon.<br>Realistic action analysis. |
| ViSOR -Video Surveillance online repository 2005 | Video sequences taken from a real surveillance setup, University of Modena and Reggio Emilia campus monitoring. | Multi camera systems composed by 8 different surveillance cameras with a non-overlapped field of views. |
| IXMAS 2006 | Scratching head, sitting down, getting up, throwing, turning around, walking, waving, punching, kicking, pointing, picking up, , nothing, checking wrist watch, crossing arms. | Multi-view, captured using 5 cameras, controlled conditions (indoor), Realistic. |
| Kinect 3D Active (K3Da) 2015 | Participants performed standardized tests, including the Short Physical Performance, Timed-Up-And-Go, vertical jump and other balance assessments, which were recorded using depth sensor technology. | Motions collected from young and older men and women ranging in age from 18 - 81 years. |

activities performed by 11 subjects 3 times each. The actors freely choose position and orientation. The details of the actions considered and the capturing environment are as shown in table 2.2.

### 2.6.6. Kinect 3D Active (K3DA)

Kinect 3D Active considers the clinically-relevant motions dataset, which is prepared using the Microsoft Kinect One sensor. Dataset is released to the community as an open source solution for benchmarking detection, quantification and recognition algorithms. The dataset includes motions collected from young and older men and women ranging in age from 18 - 81 years. Participants performed actions like Timed-Up-And-Go, vertical jump and other balance assessments which were recorded using depth sensor technology and extracted to generate motion capture data, sampled at 30 frames-per-second.

## 2.7. Challenges Identified in Domain and Need of More Research

The recognition of human action from video content is a significant area of research in computer vision, and has shown considerable progress in the domain. The paper discussed different representation and recognition strategies classified according to the complexity of action. Hierarchical approaches have shown great success for the recognition of complicated actions and interactions. Techniques like bag-of-words, and HMM that have shown success in speech and text recognition are successfully applied for action recognition. There is the need to cope with the challenges in segmentation, modelling, and occlusion handling. The system requires robust solutions, which can help in product design. Some of the challenges in the domain are discussed in the following section, and it will provide a future direction for novice researchers in the domain.

### 2.7.1. Designing Invariant System

Designing an invariant system is one of the major challenges as same action class shows the wide variability in the features. From the reported literature, in-class variability occurs due to three main factors: (a) Action execution rate, (b) Human anthropometry, and (c) Camera view point. The humanoid image model representations are highly sensitive to in class variability than the humanoid body model.

- **Action Execution Rate**

  Action execution rate for a particular action may vary with time for the same person, and there can be a difference in the action execution rate of different persons. DTW is applied by many researchers to overcome variability due to the execution rate. Pham et al. applied DTW and voting algorithms on the 3D human skeletal models [66]. They have claimed that their methods have outperformed the previous methods in both recognition accuracy, and computational complexity.

Variability in the execution rate may appear due to an expert level in performing an action, or it may vary with the situation. Dance steps or exercises in rehabilitation may show a significant difference in the execution rate due to an expert level in a performing an action. Whereas, in a running action, situations like a street dog running behind, compels a person to run faster.

Even though these approaches are successful in lab experiments, they may fail for real world practical problems.

- **Human Anthropometry**

  Anthropometric (body measurement) invariance is one of the important parameters to be considered for a robust system design. Anthropometric variances are seen due to change in size, shape, gender, and other similar parameters. In applications like surveillance anthropometry, it may not be very significant for action recognition, but it plays a major role in motion analysis applications. This problem has received meagre attention of the researchers.

- **Camera View Point**

  Camera view point plays a very important role for applications like surveillance. Successful approaches for a single view point miserably, fail for other view-points. Some researchers have proposed systems for view-invariant recognition of motion. Rao and Shah have designed the view invariant representation and recognition method using the spatiotemporal curvature of motion trajectories [40]. The incremental learning approach is used for training the system for a given action captured from the different viewpoints for different people. But, the proposed methods didn't show any success in more general situations.

### 2.7.2. Intention Reasoning

Intention reasoning plays a very important role in security applications. For most of the cases, the system is trained for specific actions, and may fail for different unpredicted actions. For the detection of fighting, if the system is trained for kicking and punching, it may fail to detect hitting with an object in fighting even though it is a part of fighting. Today's systems may fail to identify the difference between Karate practice and actual fighting due to ignorance of the intention reasoning parameter.

Hence, there is a strong need of designing robust generalized systems with intention reasoning.

### 2.7.3. Providing Product for Real World Problems

Various available algorithms for human motion representation, and recognition are mainly, driven by specific applications or datasets. Many researchers and organizations are actively involved in the domain, and have provided a variety of datasets. Chaquet et al. have reported almost 68 datasets with a variety of motions including complex behavior [65]. In order to design and deploy vision based systems for various surveillance, and control and analysis applications in real time environment, there is the need of a rigorously prepared, standardized common dataset to assess and compare the algorithmic performances. This will lead the Computer Vision community to provide a robust solution for real world problems in various applications.

## 2.8. Summary of Survey

The recognition of human action from video content is a significant area of research in computer vision, and has shown considerable progress in the domain. The chapter discussed different representation and recognition strategies classified according to the complexity of action. Hierarchical approaches have shown great success for the recognition of complicated actions and interactions. Techniques like bag-of-words, and HMM that have shown success in speech and text recognition are successfully applied for action recognition. Advances in fields like Artificial Intelligence and Machine Learning need to be applied for human motion recognition. Now, with efforts of the research community in human motion recognition, products with intelligent camera systems for social and commercial applications should be made available in market.

Applications like traditional dances, sports, exercise require body movements to be done in a specific manner. This demands assessment of correctness of performed movements by practitioner, specifically for the novice practitioner. Corrective suggestions by expert need to be provided for incorrect actions and this part of domain caught attention of very few researchers. So, there is intense need to develop such kind of robust systems. Major challenges in these kind of system is in-class variance due to human anthropometry, action execution speed, and camera view point, sensitivity to clothing and light intensity, occlusion due to body parts etc.

# 3. MODEL DESIGN AND ALGORITHMS FOR VISION BASED AMENDMENT IN HUMAN ACTION WITH ANTHROPOMETRIC INVARIANCE

*"You never fail until you stop trying"*

*-Albert Einstein*

Some applications like traditional dances, sports, exercises require body movements to be done in a specific manner. This demands assessment of correctness of performed movements by practitioner, specifically for the novice practitioner. Corrective suggestions by expert need to be provided for incorrect actions. Main goal of research work is to design vision based system to assess correctness of performed actions and suggest amendments in incorrect actions. Designed system is named as *e-YogaGuru* and tested on *Yogāsana* dataset. Detailed design of system along with algorithms and created dataset is discussed in this chapter

## 3.1. Dataset Creation

Main motivation behind consideration of *Yogāsana* as dataset for suggestion of amendment in human action is its ability to soothe the nerves, and calms the brain to make the mind fresh and relaxed, and the body healthy and active [67]. In modern age, researchers have experimented and proved that, regular practice of *Yogāsana* gives tremendous benefits in psychological and physiological disorders like blood pressure, diabetes, immune conditions, muscle strength, weight loss, and depression [23] - [25]. On 11 December, 2014, the 193-member of United Nations General Assembly approved by consensus, a resolution establishing 21 June as "International Day of *Yoga*". The first International Day of *Yoga* was observed over the world on 21 June, 2015.

### 3.1.1. *Yogāsana*

*Yoga* is one of the six orthodox systems of Indian philosophy. The word *Yoga* is derived from the Sanskrit root *Yuj*, which means to join, bind, yoke, or union. Twenty-ninth Sutra of the second book of *Patañjali* gave concept of '*Ashtānga Yoga*' (Eight-Limbed

*Yoga*) [67]. Eight limbs or stages of *Yoga* for quest of soul are *Yama*, *Niyama*, *Āsana*, *Prāṇāyāma*, *Pratyāhāra*, *Dharanā*, *Dhyāna*, *and Samādhi*. The person who practice *Yoga* is called *Yogi* or *Yogini*. First two limbs *Yama* and *Niyama* control the passion and emotion of *Yogi* and keep him in harmony with his fellow man. *Āsana* is third limb of '*Ashtānga Yoga*' that means posture. *Āsana* keep the body healthy and strong and in harmony with nature. The first three stages are known as the outward quests (*Bahirang Sādhanā*). *Prāṇāyāma and Pratyāhāra* are known as the inner quests (*Antarang Sādhanā). Dharanā, Dhyāna,* and *Samādhi* are known as quests of the soul (*Antarātmā Sādhanā*). In our research we have considered only third limb of *Yoga* i.e. *Yogāsana.*

*Yogāsana* can be divided into seven categories depending on the posture (*Sthiti*) used while performing it. Categories are standing, sitting, forward bending, back bending, twisting, and supine as shown in figure 3.1.

This research considers the *Yogāsana* with the standing and the forward bending poses. Five *Yogāsana* with standing posture are *Samasthiti Tādāsana* (Mountain Pose), *Ūrdhva Hastāsana (Upward Salute), Vīrabhadrāsana-II* (Warrior Pose-II), *Vṛukṣāsana* (Tree Pose), and *Ūrdhva Baddhanguliyāsana* (Upward bound hands pose). Forward Bending *Yogāsana* are *Utkaṭāsana* (Chair pose), and *Ardhauttānāsana*. *Yogāsana* for all standing poses are captured from front view, *Ardhauttānāsana* is captured from side view and *Utkaṭāsana* is captured from 40° to 45° inclination with sensor. Details of this *Yogāsana* dataset are explained in subsequent section.

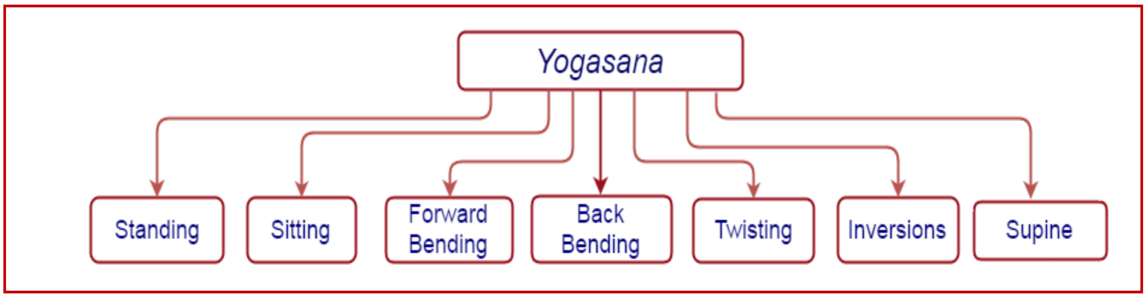



*Figure 3.1: Types of Yogāsana*

### I. *Samasthiti Tādāsana* (Mountain Posture)

*Samasthiti Tādāsana* is the foundation for other *Yogāsana*. Practicing it gives rise to a sense of firmness, strength, stillness, and steadiness [67]. Practicing it treats depression, improves incorrect posture, and strengthens the knee joints. Distinct posture sequence identified for this *Āsana* is: (a) *Samasthiti,* (b) body up with balancing body weight on toes, (c) *Samasthiti.* It is shown in figure 3.2 (a) - (c). *Samasthiti* posture as shown in figure 3.2 (a), is basic posture for all standing *Āsana*. Each standing *Āsana* starts and ends at *Samasthiti*. In *Samasthiti Tādāsana*, after standing firmly, lift body up by balancing your body weight on toes. This achieved posture needs to be hold for at least 10-30 seconds for novice practitioner. The hold time depends with different factors such as expert level in performing *Āsana,* age, health state etc. After that regain the original posture i.e. *Samasthiti*.

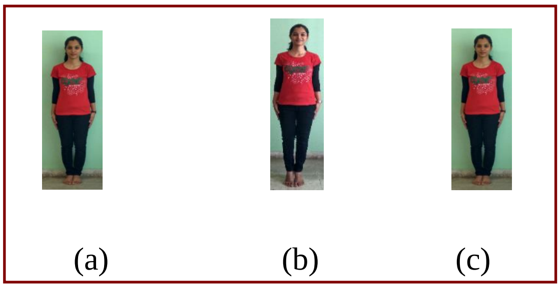



*Figure 3.2: Distinct postures sequence in Samasthiti Tādāsana*

**II. *Ūrdhva Hāstāsana* (Mountain Pose with hands stretched up)**

This is variation of the mountain pose. Practicing it provide benefits such as toning and stimulation of the abdomen, pelvis, torso, and back, relieves arthritis, boosts confidence along with benefits of *Samasthiti Tādāsana* [67]. Distinct posture sequence identified for this *Āsana* is: (a) *Samasthiti*, (b) hands apart, (c) hands up with palms facing each other, (d) stand on toes with hands up, (e) stand on feet, (f) hands apart, (g) hands down

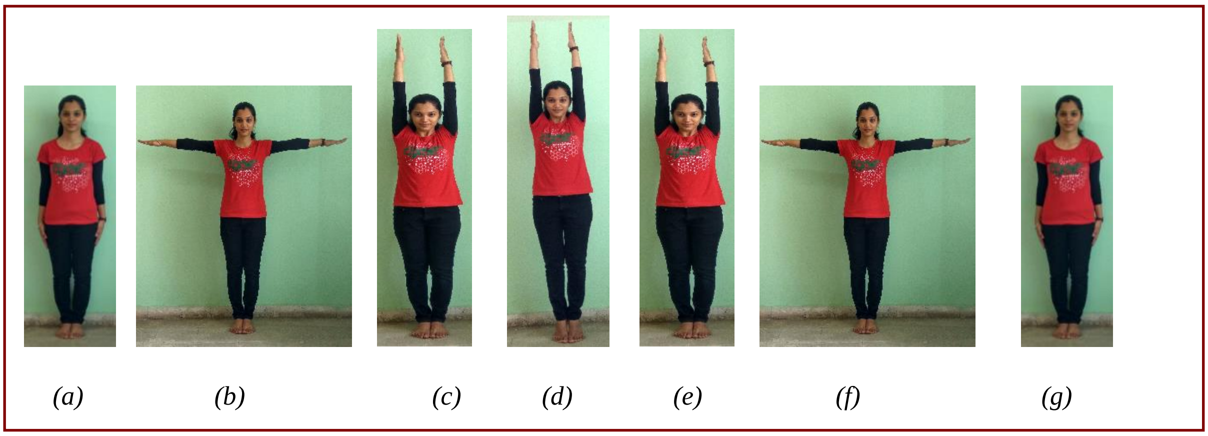



*Figure 3.3: Distinct posture sequence in Ūrdhva Hāstāsana*

as shown in figure 3.3 (a)-(g). Here, pose shown in figure 3.3 (d) standing on toes with hands up and palms facing each other is main posture in *Ūrdhva Hāstāsana* and needs to be hold for at least 10 - 30 seconds for novice practitioner.

### III. *Ūrdhva Baddhanguliyāsana* (Mountain Pose with bind hands stretched up)

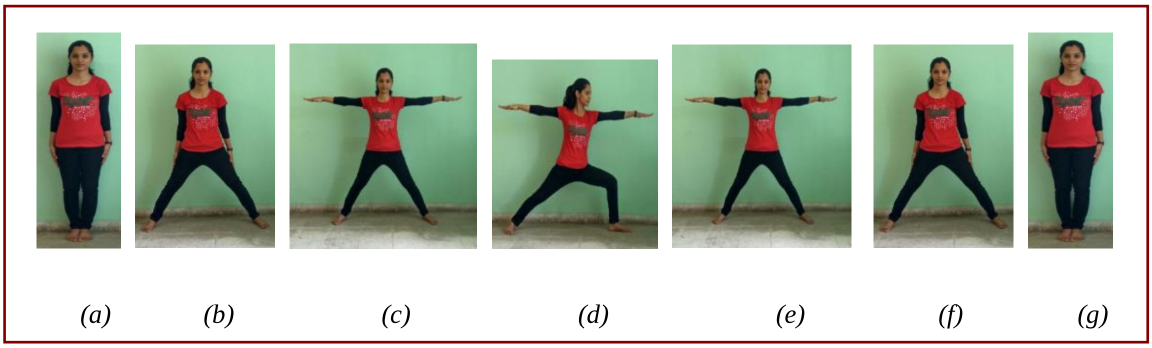
This *Āsana* is warm up pose that aligns vertebral column and protects spinal muscles and nerves. Practicing it creates space between vertebrae and prepares the spine for further stretching and flexing. Distinct posture sequence identified for this *Āsana* is: (a) *Samasthiti*, (b) hands apart, (c) hands up with fingers of both hands bind together, (d) stand on toes with hands up, (e) stand on feet, (f) hands apart, (g) hands down as shown in figure 3.4 (a) – (g). Here, standing on toes with both hands bind together and body up posture as shown in figure 3.4 (d) needs to be hold for at least 10-30 seconds by novice practitioner.

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

*Figure 3.4: Distinct posture sequence in Ūrdhva Baddhanguliyāsana*

### IV. *Vīrabhadrāsana-II* (Warrior Pose-II)

Regular practice of *Vīrabhadrāsana-II* helps to develop strength and endurance. Relieves lower backache and also, reduces fats around heap. It relives cramp in the calf and thigh muscles, tones the abdominal organs. Distinct posture sequence identified for this *Āsana* is: (a) *Samasthiti*, (b) legs apart, (c) hands apart, (d) knee bend, (e) knee straight, (f) hands down, (g) legs near. Pictorial representation of posture sequence is shown in figure 3.5 (a) – (g). Knee bend posture has two variants: knee bend with left leg and knee bend with right leg.



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

*Figure 3.5: Distinct posture sequence in Vīrabhadrāsana-II*

### V. *Vṛukṣāsana* (Tree Pose)

*Vṛukṣāsana* improves balance and stability in the legs. Strengthens the ligaments and tendon of the feet. Strengthens and tones the entire standing leg, up to the buttocks. Distinct posture sequence identified for this *Āsana* is: (a) *Samasthiti*, (b) Knee bend and foot keep on thigh, (c) *Namaskār* pose with hands, (d) *Namaskār* hands up, (e) *Namaskār* hands down(f) hands down, (g) legs down. Knee bend posture has two variants: knee bend with left leg and knee bend with right leg. Pictorial representation of posture sequence knee bend with left leg is shown in figure 3.6 (A) (a) – (g) and with

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

*Figure 3.6 (A): Distinct posture sequence in Vṛukṣāsana (With left leg)*

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

*Figure 3.6 (B) Distinct posture sequence in Vṛukṣāsana (With right leg)* right leg in figure 3.6 (B) (a) – (g). Main posture i.e. 3.6 (A) (d) and 3.6 (B) (d) needs to be hold for at least 10 – 30 seconds.

### VI. *Ardhauttānāsana*

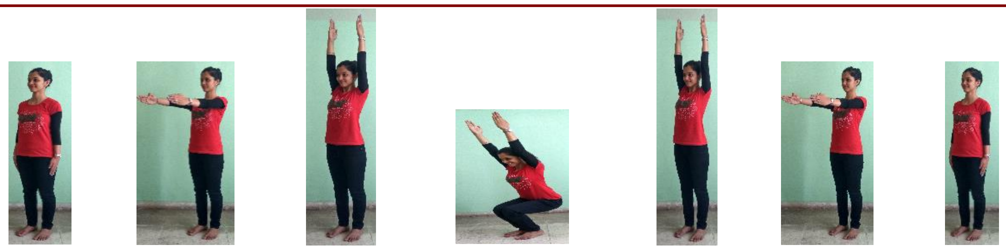
Regular practice of *Ardhauttānāsana* tones the leg muscles excellently. It strengthens hip flexors, ankles, calves, and back and stretches chest and shoulders. Furthermore, it reduces symptoms of flat feet and stimulates the heart, diaphragm, and abdominal organs. Distinct posture sequence identified for this *Āsana* is: (a) *Samasthiti*, (b) Hands Front, (c) Hands Up, (d) *Ardhauttānāsana* pose (e) Body straight, (f) Hands Front, (g) *Samasthiti*. Pictorial representation of posture sequence in *Ardhauttānāsana* is shown in figure 3.7 (a) – (g).

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

*Figure 3.7: Distinct posture sequence in Ardhauttānāsana*

### VII. *Utkaṭāsana*

Regular practice of *Utkaṭāsana* stretches the hips, hamstrings, and calves. Strengthens the thighs and knees, keeps practitioner's spine strong and flexible. It reduces stress, anxiety, depression, and fatigue and relieves tension in the spine, neck, and back and also activates the abdominal muscle Distinct posture sequence identified for this *Āsana* is: (a) *Samasthiti*, (b) Hands Front, (c) Hands Up, (d) *Utkaṭāsana pose*, (e) Body straight, (f) Hands front, (g) *Samasthiti* as shown in figure 3.8 (a)-(g). Here, *Utkaṭāsana* main posture is shown in figure 3.8 (d) and needs to be hold for at least 10 – 30 seconds.



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

*Figure 3.8: Distinct postures considered in Utkaṭāsana*

### 3.1.2. *Yogāsana* Dataset Creation

Nowadays, technological advancement in Computer Vision domain provide sensors that track the people in captured scenario and provide features for further processing. This assist researchers to concentrate more on high level vision problems, as low level vision steps human tracking and feature extraction are performed by sensor. Designed system e-*YogaGuru* uses Kinect sensor for dataset creation and kinematic data stream obtained in terms of body joint position is stored in .csv file for further processing. Kinect Xbox 1 provides additional four joint positions as compared to Kinect Xbox 360 left and right hand tip and thumb and spine mid are extra joint positions detected. This helped to get all joint points clearly in *Ūrdhva Baddhanguliyāsana*, as it requires details of bind hands position.

Practitioner needs to stand within a practical range defined by the Kinect. Distance between camera and practitioner is kept at 180 cm to 200 cm and camera is present at height 87 cm from ground for *Yogāsana* dataset creation. C apturing environment is as shown in figure 3.9 (a) for Kinect Xbox 360 and 3.9 (b) for Kinect Xbox 1.0. Dataset on seven mentioned *Yogāsana* is prepared in the specified environment. Twenty-five practitioners performed three *Yogāsana i.e. Samasthiti Tādāsana*, *Ūrdhva Hāstāsana and Vīrabhadrāsana -II* captured using Kinect 360. Total 750 video sequences (25 practitioner X 3 *Āsana* X 10 repetitions) are captured using Kinect 360. Nine practitioners performed six *Yogāsana* (all except *Samasthiti Tādāsana*) captured using Kinect Xbox 1.0. Total 540 video sequences (9 practitioner X 6 *Āsana* X 10 repetitions) are captured using Kinect 1.0. Total Number of video sequences using both sensors are 1290 (i.e. 750 + 540) of average length 1800 frames. Out of these in 645 video sequences *Yogāsana* is done correctly and in another 645 video sequences has

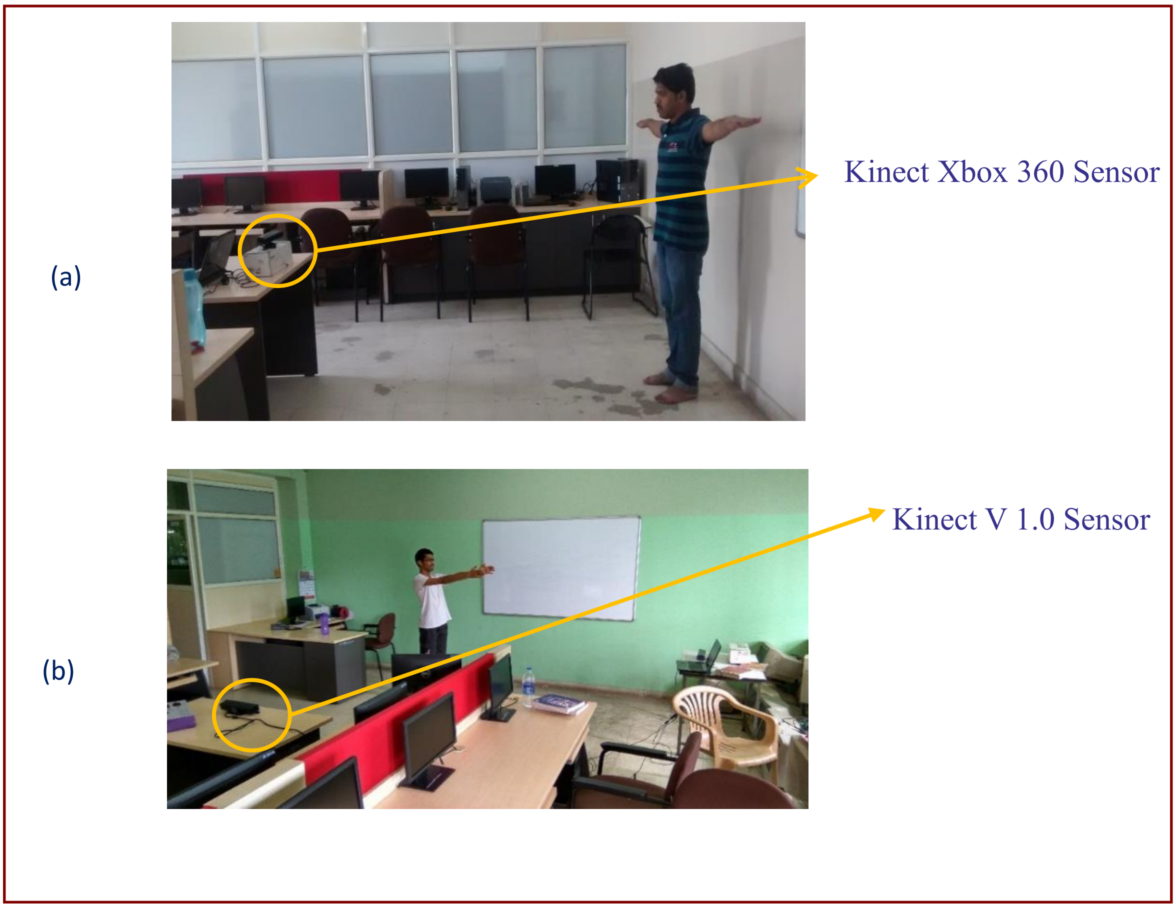



*Figure 3.9 Data capturing environment (a) Using Kinect 360, (b) Using Kinect 1*

erroneous *Yogāsana* postures. Total frames considered for recognition are approximately 23, 22,000 frames. Total duration of video sequences is 21 hours 50 minutes.

### 3.1.3. Feature Selection

After tracking human in scene, body feature extraction is important step to represent body parameters and body movement. Human posture is represented using either humanoid image model or using humanoid body model as explained in section 2.4. To assess correctness of *Āsana*, human posture needs to be represented efficiently either by humanoid body model or humanoid image model. Experimentation is performed on *Bharatnāṭyam Adavu* recognition system using image features and *Yogāsana* recognition system using kinematic features. From experimentation we conclude that humanoid body model is advantageous over humanoid image model. So, humanoid body model is considered for human posture representation. Human skeleton is represented using stick figure and human joint points are used as features. Stick figure representation for all seven *Yogāsana* are shown in figure 3.10 (A) – (G).

It is observed that for some *Yogāsana* side view provides more accurate feature information than front view. *Samasthiti Tādāsana*, *Ūrdhva Hāstāsana, Vīrabhadrāsana-II*, *Vṛukṣāsana*, *Ūrdhva Baddhanguliyāsana* are captured from front view, *Ardhauttānāsana*

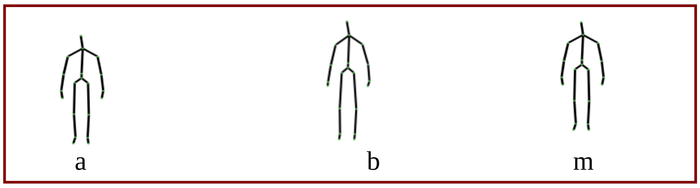



Figure 3.10 (A): Stick figure for posture sequence in *Samasthiti Tādāsana*

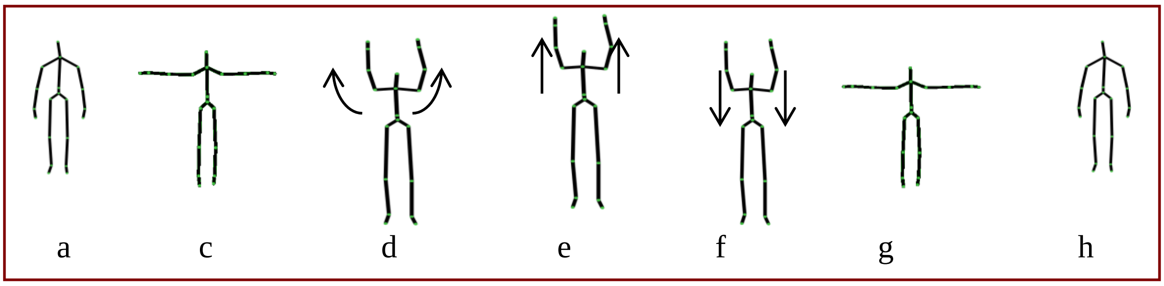



Figure 3.10 (B): Stick figure for posture sequence in *Ūrdhva Hāstāsana*

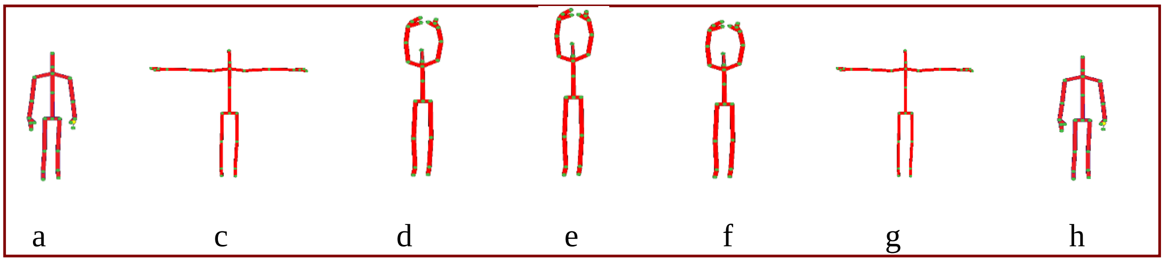



*Figure 3.10 (C): Stick figure for posture sequence in Ūrdhva Baddhanguliyāsana*

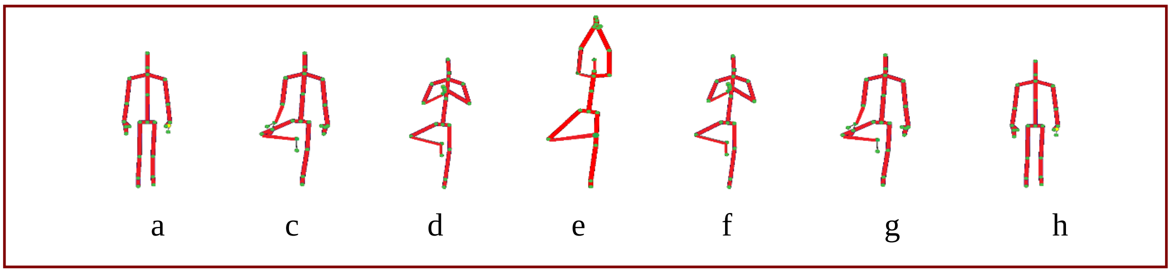



*Figure 3.10 (D): Stick figure for posture sequence in Vṛukṣāsana (With right leg)*

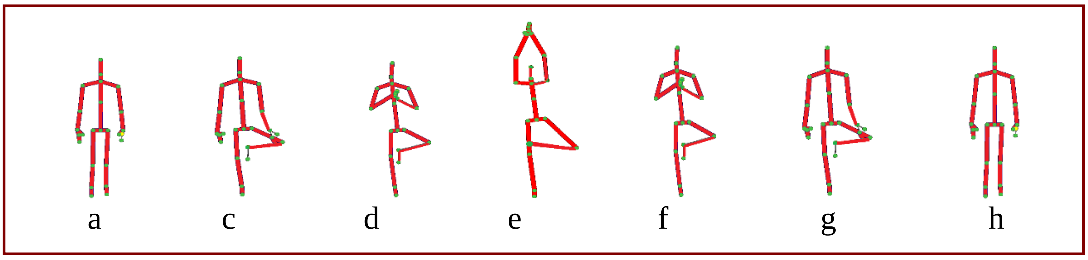



*Figure 3.10 (E): Stick figure for posture sequence in Vṛukṣāsana (With left leg)*

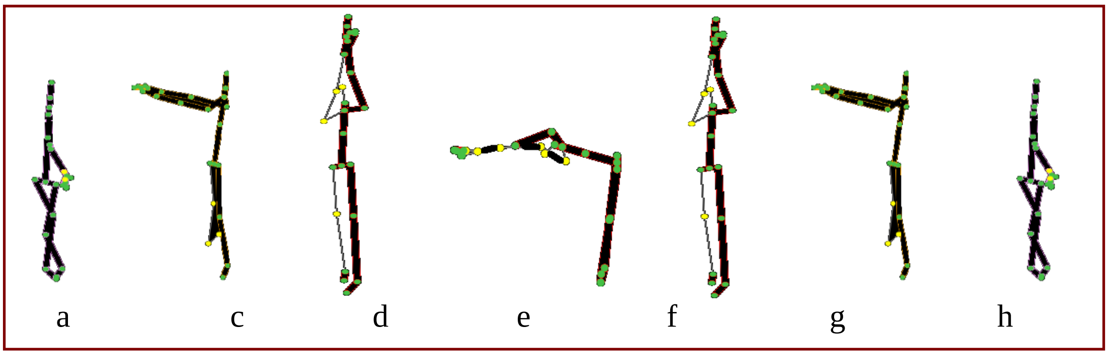



*Figure 3.10 (F): Stick figure for posture sequence in Ardhauttānāsana*

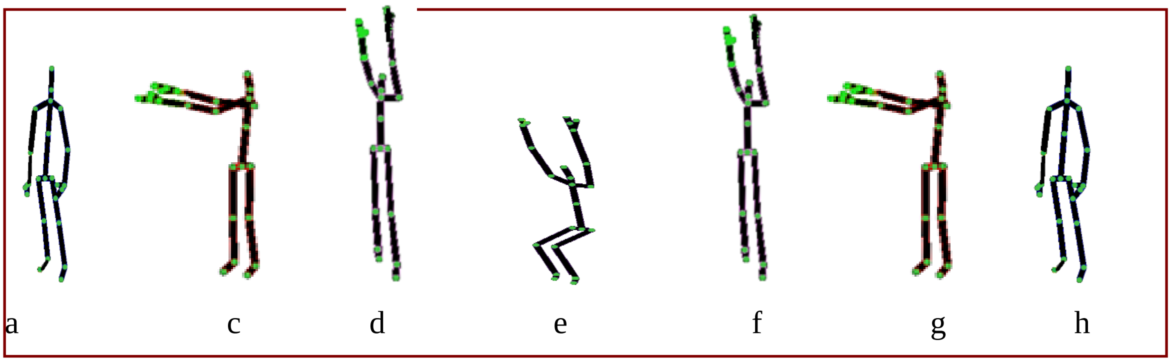



*Figure 3.10 (G): Stick figure for posture sequence in Utkaṭāsana*

is captured from side view as shown in figure 3.7 and figure 3.10 (F) and *Utkaṭāsana* from $40^o$- $45^o$ inclined view as shown in figure 3.8 and figure 3.10 (G).

After representation of features, human posture needs to be identified using efficient recognition method. Motion trajectories template matching approaches can be used for recognition of performed action. However, there is a vast variation in human anthropometry and ultimately in joint positions' values. This will lead to difference in motion trajectories for same movement performed by different persons. The angle features can be used for representation of each action, as they are insensitive to change in length.

For dataset of five *Yogāsana* captured from front view, we have taken ten body angles as shown in figure 3.11. Blue color shows joint positions and red angle values. Total number of angles and angle position may vary with considered action set for recognition.

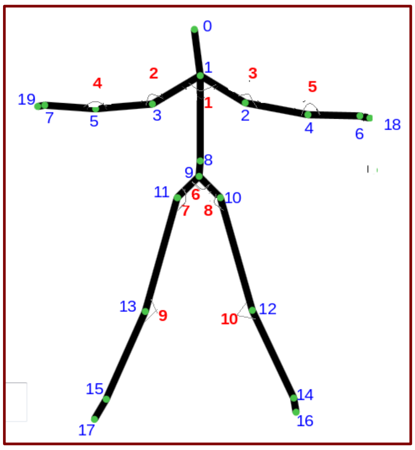



*Figure 3.11: Angles for representation of Yogāsana*

Angle details for prepared *Yogāsana* dataset are,

∠1 - Angle made by right shoulder, left shoulder and shoulder center joint.

∠2 - Angle made by right shoulder, shoulder center joint and right elbow.

∠3 - Angle made by left shoulder, shoulder center joint and left elbow.

∠4 - Angle made by right shoulder, right elbow and right wrist joint

∠5- Angle made by left shoulder, left elbow and left wrist joints

∠6 - Angle made by right hip, left hip and hip center joints.

∠7 - Angle made by right hip, hip center joints and point obtained from X-value of spine and Y-value of right knee.

∠8 - Angle made by left hip, hip center joints and left knee

∠9 - Angle made by right hip, right knee and right ankle joints

∠10 - Angle made by left hip, left knee and left ankle joints

*Ardhauttānāsana* and *Utkaṭāsana* are captured from side view and some more angles such as angle between legs and spine needs to considered.

From given three point values angle can be computed using triangle geometry as shown in figure 3.12.

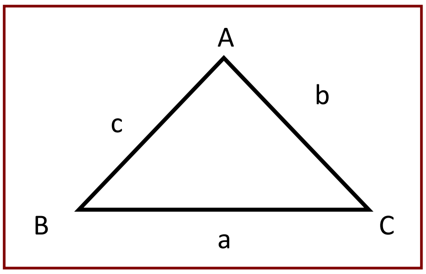



*Figure 3.12: Triangle for angle computation*

Consider joint positions *A* $(X_1, Y_1)$, *B* $(X_2, Y_2)$, *C* $(X_3, Y_3)$. We can compute values of $\angle A$, $\angle B$ and $\angle C$ as follows,

Distances a, b and c are computed using distan-ce formula,

$$a = \sqrt{(X_3 - X_2)^2 + (Y_3 - Y_2)^2}$$

$$b = \sqrt{(X_3 - X_1)^2 + (Y_3 - Y_1)^2}$$

$$c = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

*∠A, ∠B, and ∠C are computed using,*

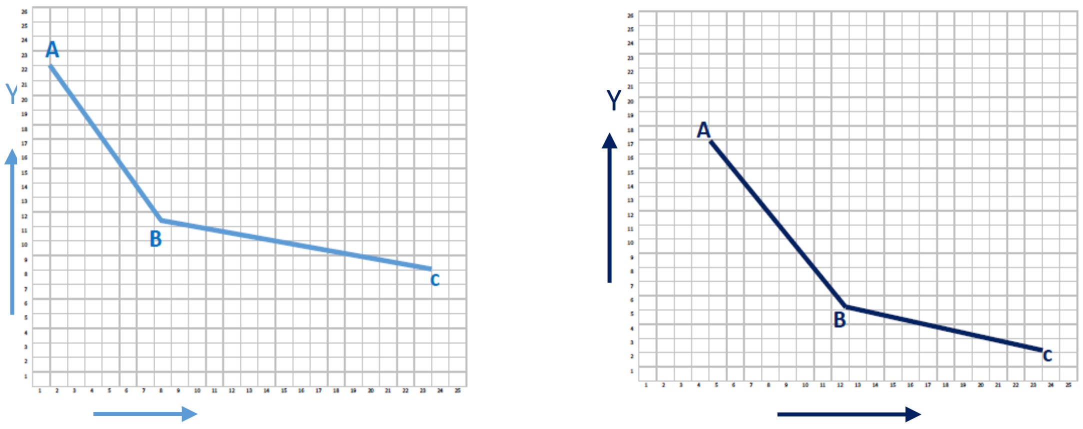
$$\angle B = cos^{-1}\left(\frac{a^2+c^2-b^2}{2\times a\times c}\right),$$

$$\angle C = sin^{-1}\left(\frac{c \times sin B}{b}\right)$$

$$\angle A = 180 - (B + C)$$

Some angle positions are shown in figure 3.13 and it is observed that few angles like $\angle 2$ *and* $\angle 3$, measured anticlockwise direction possess value more than *90º* for Samasthiti position, more than 180º for hands apart and more than *270º* for hands up position. It is observed that using triangle geometry is not suitable for angles more than

*180°* as it always computes interior angle and cannot compute reflex angle. So, body angles are computed using vector geometry as explained in equation (1).




**(a)** **(b)**

*Figure 3.13: Vector geometry*

Theoretical analysis consideration of angle features over joint position for representation of human posture is done in subsequent discussion.

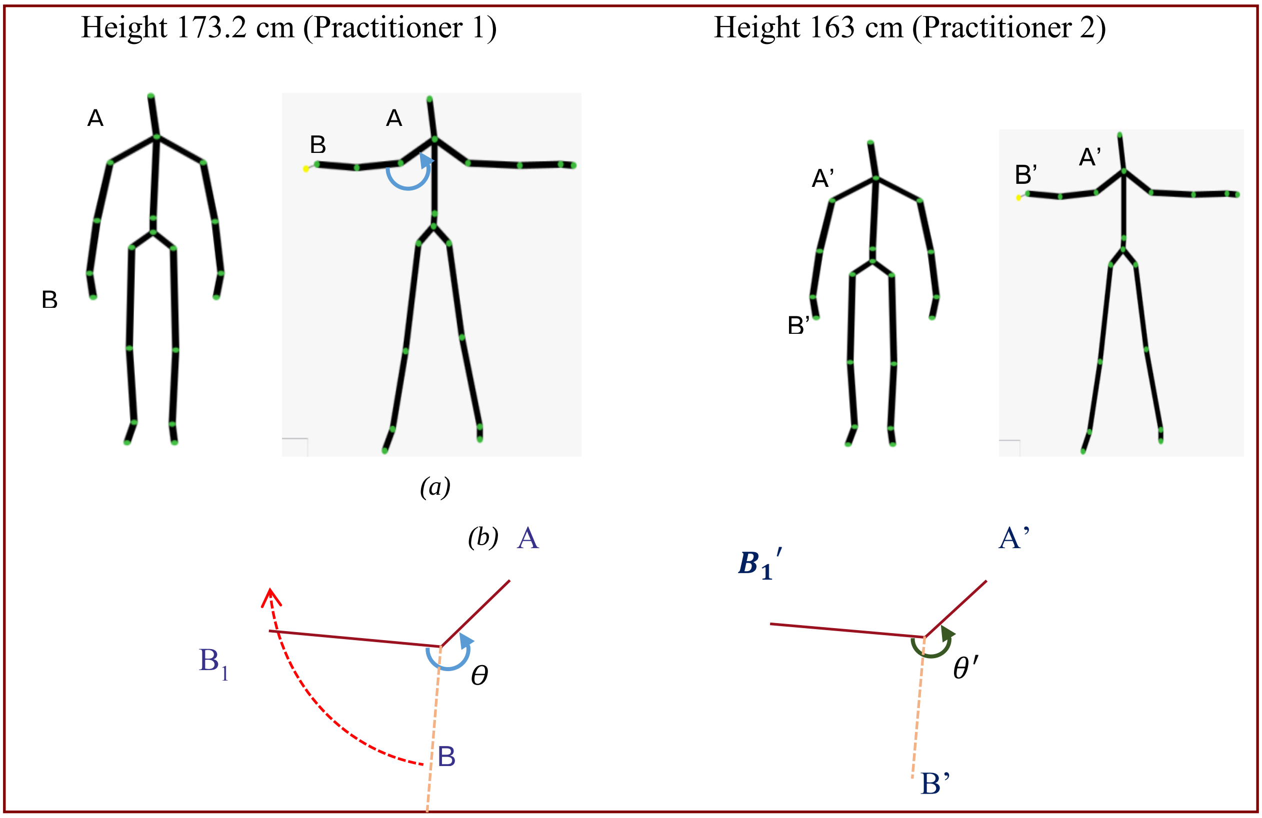
Figure 3.13 (a) and (b) shows two vector geometries considered for computation of $\measuredangle B$. Co-ordinate values in figure 3.13 (a) are A (1, 22), B (8, 12) and C (20, 9) and co-ordinates values in figure 3.13 (b) are A (4, 17), B (12, 5) and C (24, 2). Segment AB and segment BC lie at different locations on grid and length of line segment AB is different in both geometries. $l(AB) = 14.42$ in figure 3.13 (a) and $l(AB) = 12.20$ in figure 3.13 (b). Angle values are computed using,

$$\left.\begin{aligned} &dot = (X_1-X_2)*(X_3-X_2) + (Y_1-Y_2) * (Y_3-Y_2) \\ &det = (X_1-X_2)*(Y_3-Y_2) - (Y_1-Y_2) * (X_3-X_2) \\ &ang = atan2(det, dot) \\ &ang1 = ang * 180 * 7/22 \\ &\quad if(ang1 < 0)\ then\ ang1 = -ang \\ &\qquad\qquad else\ ang = 360 - ang \end{aligned}\right\} \quad (1)$$

$\measuredangle B$ =221.0277 for figure 3.13 (a) and $\measuredangle B$ =221.1107 for figure 3.13 (b). If we observe the co-ordinate values and angle values of figure 3.13 (a) and (b), even though there is difference in co-ordinate values angle values are almost similar. This shows that consideration of angle feature will give more robust features than co-ordinate values. D theory is experimented on actual data captured using Kinect sensor on nine participants with different height and weight for hands apart posture. Table 3.1 shows data for nine *Yogāsana* practitioner, *(X, Y)* values for three joint points i.e. shoulder center, shoulder right and elbow right, computer angle value at right shoulder and Body Mass Index (BMI). Actual angle and positions are explained in figure 3.14. BMI is taken as parameter for representation of anthropometry. Joint data is obtained using Kinect sensor. By observing joint position and corresponding angle values, definitely angle features are more robust than joint values.

Selection of angle feature for representation of body parameters gives following advantages:

**Reduced Feature vector**: Feature vector is reduced from 50 (40 for Kinect V1.0) dimensional to 10 dimensional and more accurate and discriminative features are considered.




(c)

*Figure 3.14: Angle at hands apart position*

*Table 3.1:Co-ordinates of three joint points along with computed angle and BMI*

| **Name** | **$X_1$** | **$Y_1$** | **$X_2$** | **$Y_2$** | **$X_3$** | **$Y_3$** | **Angle $\theta$** | **BMI** |
|---|---|---|---|---|---|---|---|---|
| Practitioner | 161 | 254 | 126 | 253 | 177 | 351 | 259.0376 | 18.0632 |
| Practitioner 2 | 146 | 248 | 108 | 249 | 167 | 361 | 261.0195 | 20.054 |
| Practitioner 3 | 167 | 258 | 129 | 258 | 180 | 363 | 262.9812 | 22.519 |
| Practitioner 4 | 168 | 261 | 132 | 260 | 183 | 366 | 260.3189 | 29.217 |
| Practitioner 5 | 202 | 291 | 167 | 291 | 214 | 374 | 261.8128 | 21.292 |
| Practitioner 6 | 170 | 269 | 132 | 269 | 185 | 370 | 261.5921 | 24.081 |
| Practitioner 7 | 144 | 241 | 110 | 244 | 161 | 340 | 265.3369 | 16.689 |
| Practitioner 8 | 178 | 285 | 140 | 285 | 199 | 378 | 257.317 | 28.159 |
| Practitioner 9 | 161 | 260 | 125 | 260 | 176 | 349 | 260.4734 | 23.821 |

**Anthropometric Invariance:** Motion trajectories are more sensitive to anthropometry than body angles.

But, for some postures like *Ūrdhva Hāstāsana* main posture, *Samasthiti Tādāsana* main posture, where practitioner needs to stand on toes consideration of only angle values is not sufficient and *Y*-co-ordinate of head and foot needs to be considered to find change in height of subject. So, it is observed that along with angles some joint positions need to be considered for representation of few postures. All possible distinct postures for representation of considered seven *Yogāsana* are shown in figure 3.15. Table 3.2 gives all postures and features required to be considered for recognition. Sample possible angle positions are shown in figure 3.16.

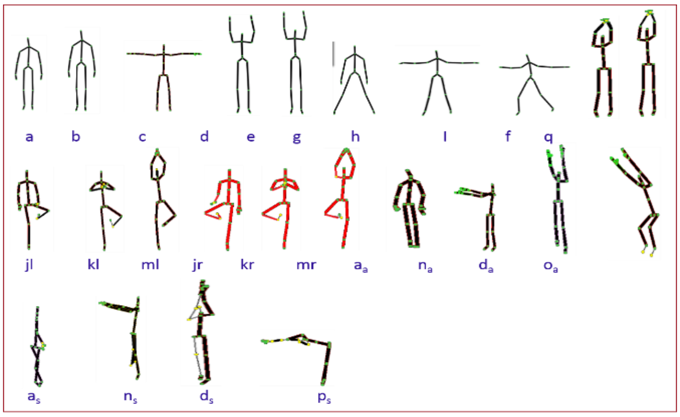



*Figure 3.15: All gesture classes considered for recognition*

*Table 3.2: Posture detail, labels used and required features for recognition*

| Sr. No. | Posture Label | Posture | Angle Feature | (X,Y) position |
|---|---|---|---|---|
| **1** | a | *Samasthiti* | Y | |
| **2** | b | *Samasthiti* body up | N | Y |
| **3** | c | *Ūrdhva Hāstāsana* hands apart | Y | |
| **4** | d | *Ūrdhva Hāstāsana* hands up | Y | |
| **5** | e | *Ūrdhva Hāstāsana* hands up and body up | N | |
| **6** | f | *Ūrdhva Baddhanguliyāsana* posture Hands up | Y | Y |
| **7** | g | *Vīrabhadrāsana-II* legs apart | Y | |
| **8** | h | *Vīrabhadrāsana-II* hands and legs apart | Y | |
| **9** | il | *Vīrabhadrāsana-II* knee bend with left leg | Y | |
| **10** | ir | *Vīrabhadrāsana-II* knee bend with left leg | Y | |
| **11** | jl | Left leg in *Vṛukṣāsana* pose | Y | |
| **12** | jr | Right leg in *Vṛukṣāsana* pose | Y | |
| **13** | kl | Left leg in *Vṛukṣāsana* pose with *Namaskār* Hands | Y | |
| **14** | kr | Right leg *Vṛukṣāsana* pose with *Namaskār* Hands | Y | |
| **15** | ml | Left leg in *Vṛukṣāsana* pose with *Namaskār* hands up | Y | |
| **16** | mr | Right leg *Vṛukṣāsana* pose with Namaskār hands | Y | |
| **17** | $a_a$ | *Samasthiti* from $35^0$ | Y | |
| **18** | $n_a$ | Hands front from $35^0$ | Y | |
| **19** | $d_a$ | Hands up from $35^0$ | Y | |
| **20** | $o_a$ | *Utkatāsana* | Y | |
| **21** | $a_s$ | *Samasthiti* side pose | Y | |
| **22** | $n_s$ | Hands front side pose | Y | |
| **23** | $d_s$ | Hands up side pose | Y | |
| **24** | $p_s$ | *Ardhauttānāsana* main posture | Y | |
| **25** | q | *Ūrdhva Baddhanguliyāsana* hands up, body up | Y | Y |

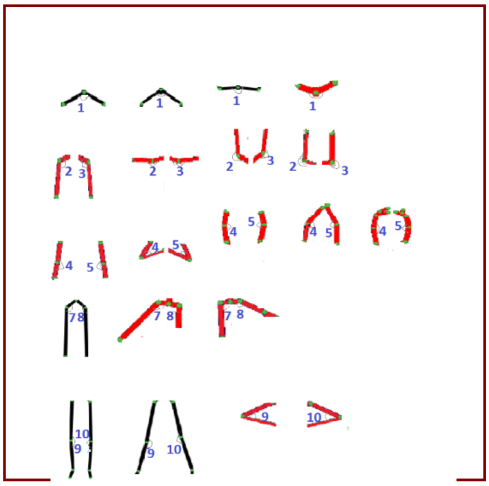



*Figure 3.16: Possible angle positions at different gestures*

## 3.2. Design of System

*Yogāsana* action dataset is comparatively long duration and simple recognition approaches are not suitable for recognition and suggestion of amendment in *Yogāsana*. Two layer hierarchical, model for recognition of *Yogāsana* and three layer hierarchical Model for suggestion of amendment in incorrect *Yogāsana* are designed. Both the models are discussed in detail in section 3.2.1 and 3.2.2.

### 3.2.1. Two Layer Hierarchical Model for Human Motion (*Yogāsana*) Recognition

Two layer hierarchical Model for recognition of *Yogāsana* identifies individual gestures from each frame using body angle and height feature at gesture layer and recognizes performed *Yogāsana* at higher layer as shown in figure 3.17. Vector geometry as discussed in section 3.1.3 is used for computation of the body angle features from joint point values. *Yogāsana* knowledge base for all postures shown in table 3.2 is created using experts' data and allowed deviation is decided from human anthropometry. Knowledge base provides minimum and maximum angle range for all *Yogāsana* postures. Experts' joint positions are recorded for few seconds and mode (most frequent values) of all the joint values are used for designing knowledge base. Human anthropometric parameters and expert level of performing *Yogāsana*, are considered for computation of Δ deviation allowed for each

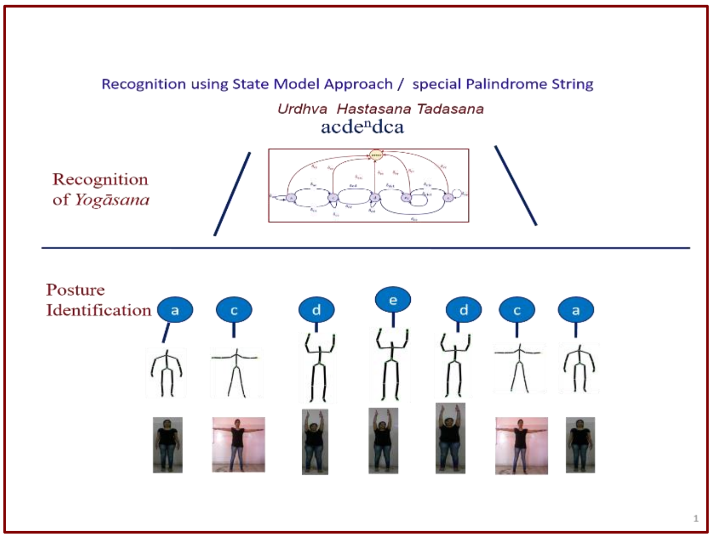



*Figure 3.17: Two layer hierarchical Model for recognition of Yogāsana*

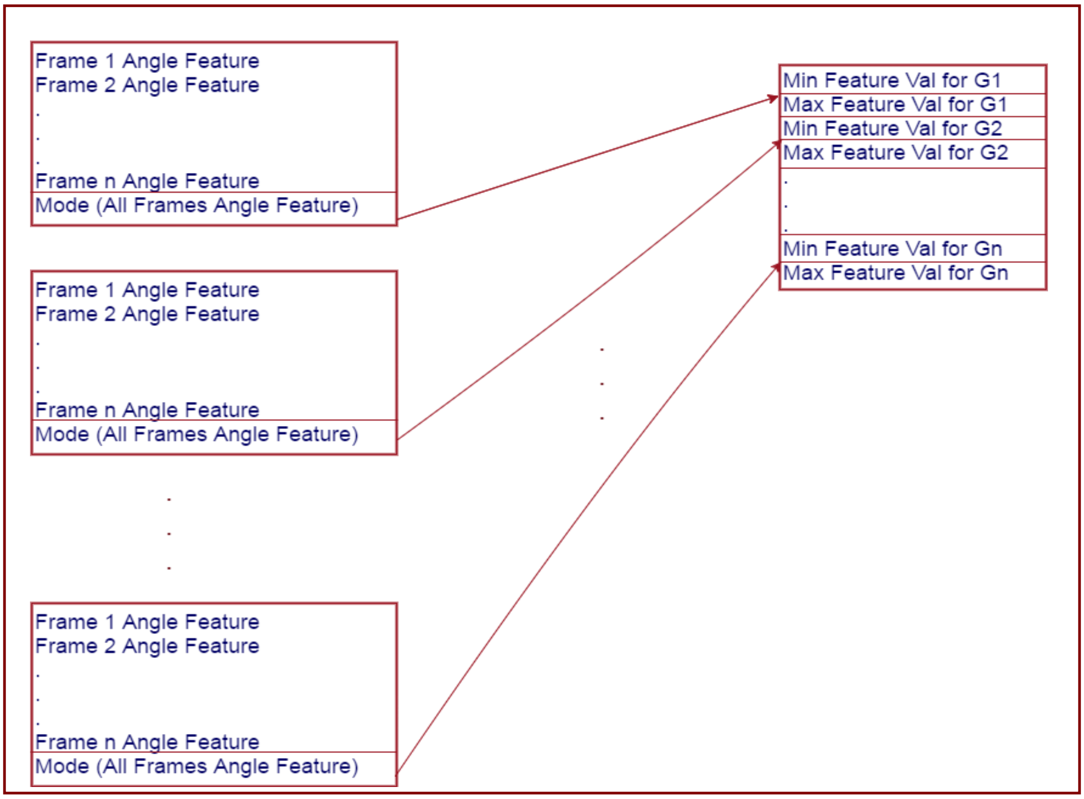



*Figure 3.18: Knowledge base creation for Yogāsana.*

considered body angle. Minimum and maximum range of each angle is decided as explained in figure 3.18. All video frames are analyzed and posture is identified using Euclidian Distance for posture feature matching as shown in figure 3.19. Identified postures are then represented using gesture label shown in table 3.2. Further, frequency count of continuous occurrence of gesture is computed and gesture sequence is provided

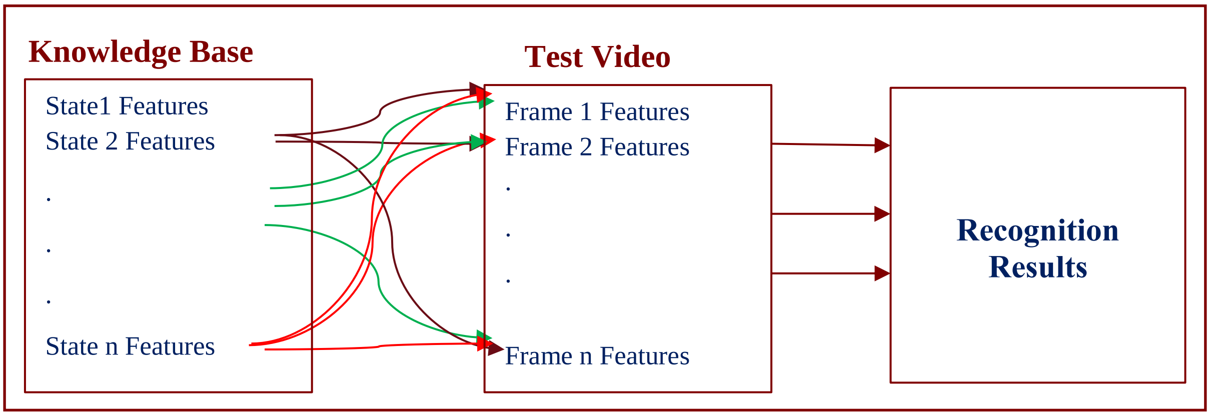



*Figure 3.19: Feature matching for recognition of Āsana*

as input to recognition layer. Recognition layer uses designed state model as discussed in section 3.3 for recognition of *Yogāsana*.

### 3.2.2. Three Layer Hierarchical System for Suggestion of Amendment

Main objective of research is to assess correctness of performed *Yogāsana* and suggest amendment for incorrect *Yogāsana*. Two layer hierarchical model identifies the *Yogāsana* from the captured data, but it is not capable of suggesting amendment. So, the third layer is added, that identifies the incorrect postures and also recommend corrective suggestion in terms of body angle. In order to suggest amendment frame-wise gesture occurrences needs to be maintained to identify and display frames with incorrect posture.

Here, posture layer identifies individual posture from each frame and provide input as posture label and frame sequence numbers. Recognition layer assess correctness of *Yogāsana*, provides feedback for correct *Yogāsana* and forward the information of incorrect *Yogāsana* to amendment layer for detailed analysis. Amendment is suggested at abstract level in terms of missing or incorrect posture name in sequence along with position. In detailed analysis incorrect body angle for erroneous posture is identified and feedback is provided in terms of frame sequence, body angle position, measured angle at that position and correct required angle range.

Two layer hierarchy uses Euclidian Distance (ED) between the summation of angle values between knowledge base created using expert's data and practitioner's data for each posture. This is not suitable for suggestion of amendment, as it fails to assess individual body angle position for correctness. For identification of error individual body angle needs to be assessed. At *Yogāsana* layer recognition is done by applying ED for individual body

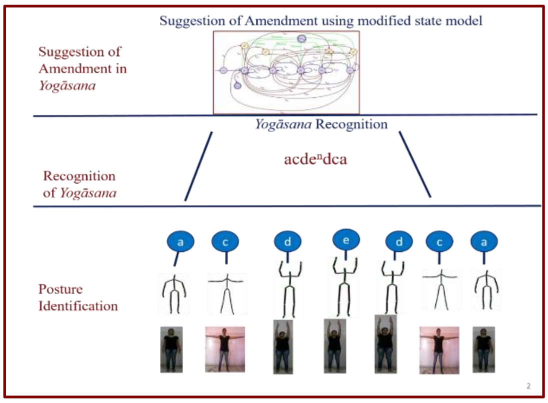



*Figure 3.20: Three layer hierarchical system for suggesting amendment*

angle. Appropriate message for correct and incorrect *Yogāsana* needs to be displayed. Details are explained in algorithm 5 and 6.

### 3.2.3. Architecture of the System

Architecture of developed system is as shown in figure 3.21. The system uses knowledge- base as reference for assessment of correctness of *Yogāsana*. Knowledge base is designed using expert's postures and by considering anthropometric parameters in consultation with trainer. Skeleton stream of expert in the form of joint points provided by Kinect is used for computation of angle values between body parts. Angles are computed using geometry explained in section 3.3.1. Knowledge base created for *Yogāsana* postures is used for recognition of practitioners' postures.

Practitioner should perform *Yogāsana* in camera range and individual posture of practitioner from each frame is identified at posture layer. Practitioner's postures are also represented using angle features and individual angles are verified for computed angle range in knowledge base. Identified postures are labeled and passed to activity layer (*Yogāsana* recognition layer). Identified postures are encoded using frequency count of each posture in continuous frames. Noise introduced during posture transition is removed. Verify correctness of *Yogāsana* using modified state transition model. At activity layer *Yogāsana* is recognized further, amendment is suggested for incorrect *Yogāsana*.

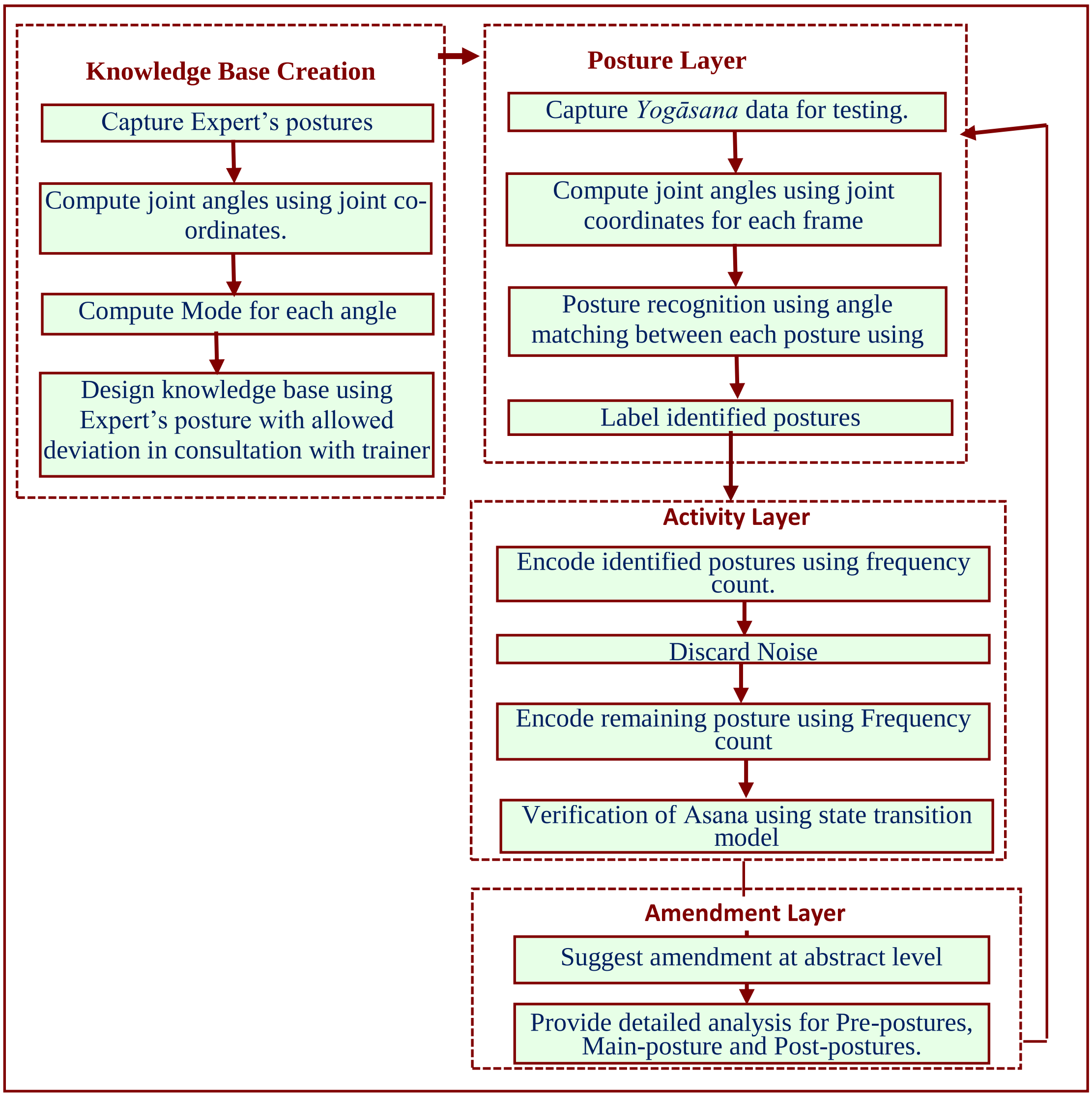



*Figure 3.21: Architecture of System for Yogāsana Recognition and amendment*

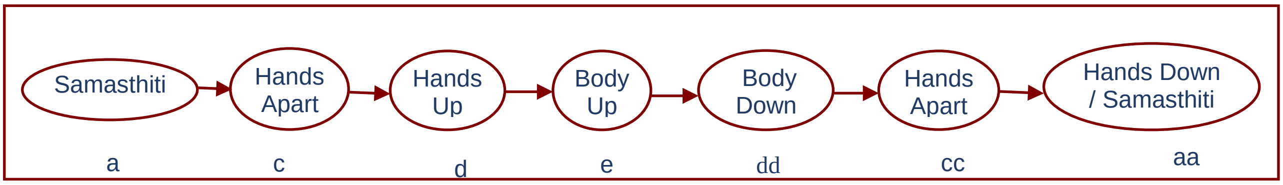
## 3.3. Design of the State Model for *Yogāsana*

Major advantage of using state model for recognition of action is invariance to action execution speed. Other recognition approaches classifies action in its class and may not help in suggestion of amendment. Using state model error at particular state is identified and feedback is provided to the practitioner. Design of system is discussed for *Ūrdhva Hāstāsana* in detail and brief description is given for all other *Yogāsana* along with design of state model for each *Yogāsana*. For each *Yogāsana* four state transition diagrams are described. *Yogāsana* is viewed as transition of posture (*sthiti*) sequence from one state to another. Here, state means distinct prominent posture in *Yogāsana*. First simple state transition diagram shows posture transitions in *Yogāsana*.

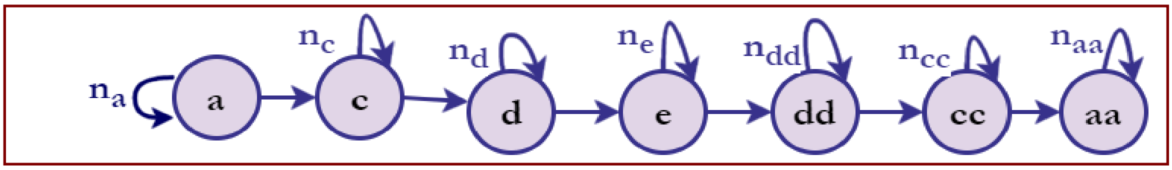
Further, identified sequence is represented symbolically in second state diagram. Third transition diagram is used for assessment of correctness of *Āsana* and forth is used for suggestion of amendment in *Yogāsana.*

### 3.3.1. *Ūrdhva Hāstāsana*

Figure 3.22 (a) shows prominent distinct postures in *Ūrdhva Hāstāsana* represented using states and arranged according to their occurrences in *Yogāsana*. Further, for simplicity states are represented using labels and modified state transition is shown in figure 3.22 (b). Practitioner will hold each posture for some amount of time. Main posture is recommended to be hold for at least 10 seconds, for new practitioner and it can be hold for more time by experts. Other postures can be hold for half to 5 seconds or more. In figure 3.22 (b) all states have a self-loop and $n_i$ represents the amount of time the posture hold. Where *i* is state name symbol. From figure 3.22 (b), it is observed that, state a and state aa have same posture i.e. *Samasthiti*. Similarly, state c is same as state cc i.e. hands apart and state d and state dd have same postures i.e. hands up. All the states of *Ūrdhva Hāstāsana* can be divided into pre-postures, main-posture and post-postures. States a-c-d is pre-posture state sequence, state e is main-posture and aa-cc-dd is post-posture state sequence. It is observed that post-posture state sequence is exactly reverse of pre-posture sequence. So, representation is further modified and similar posture states are considered only once.




(a)




*(b)*

*Figure 3.22(a): State transition sequence for Ūrdhva Hāstāsana, and (b): State transition sequence using labels and hold time for Ūrdhva Hāstāsana*

Modified state transition diagram for correctness assessment of *Ūrdhva Hāstāsana* is as shown in figure 3.23. Posture sequence is started from left-side with *Samasthiti*. For all postures other than mentioned in *Ūrdhva Hāstāsana* sequence is directed to 'error' state. Occurrence of error term insures the incorrect transition of *Yogāsana*. Main-posture i.e. e needs to be hold for at least 10 seconds. So, state $e_1$ is introduced, it has same posture features as e. Transition of state $e_1$ to e ensures that main-posture is hold for 10 seconds. Here transitions shown with blue color are correct transitions and transitions shown with brown color are incorrect transitions. Abstract description of behavior of *Yogāsana* recognition system with the example of *Ūrdhva Hāstāsana* is represented mathematically as follows,

All state model designed for *Yogāsana* and discussed in this research work starts and ends at *Samasthiti* for all correct sequences. *Samasthiti* is taken as start and end state.

*Yogāsana* with n states can be represented as,

$Yog\bar{a}sana\ = \{P_{1,}\ P_{2,}\ P_{3,}\ \dots\dots,\ P_n\}$, Where, $P_i$ is distinct posture in given *Āsana* $1 \leq i \leq n$.

Each posture $P_i$ can be represented by the sequence of the ordered pair, $\langle posture\ name, number\ of\ occurrences\ of\ posture\rangle$.

*Yogāsana Ūrdhva Hastāsana* is represented using,

$$Y_{UH} = \{\langle a, n_a\rangle, \langle c, n_c\rangle, \langle d, n_d\rangle, \langle e_1, n_{e_1}\rangle, \langle e, n_e\rangle, \langle d, n_d\rangle, \langle c, n_c\rangle, \langle a, n_a\rangle\}$$

$n_i$ – Number of frames with $i^{th}$ posture. For correct posture sequence $n_{e_1} = 290$ frames

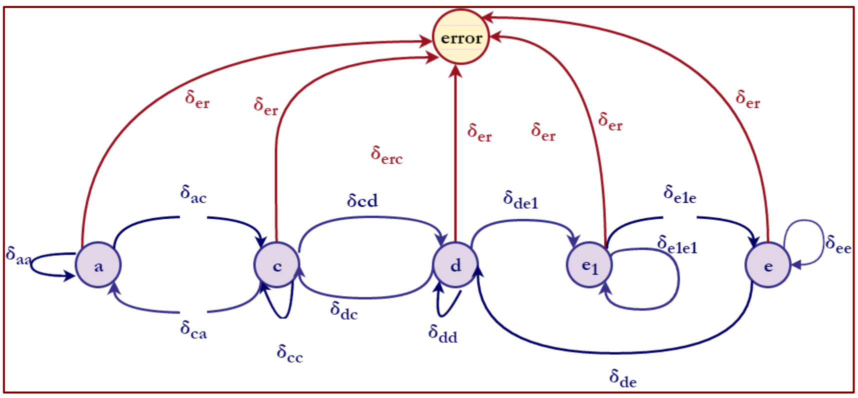



*Figure 3.23: State transition diagram for recognition of Ūrdhva Hāstāsana*

For correct posture sequence $n_{e1} = 290$, this ensure main posture hold time.

$S_{UH} -$ Represents all required distinct states present in *Ūrdhva Hāstāsana* and given by, $S_{UH} = \{a, c, d, e_1, e, error\}$

$\delta_{ij}$ and $\delta_{kk} -$ Represent input to states, while transition from state $i^{th}$ to state *j* and state *k* to state *k* respectively.

$\delta_{ij} = \langle A_{ij}, h_{ij}, SS_{ij} \rangle$

$\delta_{kk} = \langle A_{kk}, h_{kk}, SS_{kk}, t_k \rangle$

$A_{ij} = \left\{\theta_{ij1,}\ \theta_{\mathrm{ij2,}}\ \ldots,\ \theta_{\mathrm{ij10}}\right\}\quad \theta_{ij\mathrm{m}} - m^{th}\ body\ angle\ value\ for\ state\ j$

$h_{ij} = h_j - h_i,\ \ h_{ij} -\ difference\ in\ height\ while\ transition\ from\ state\ i\ to\ state\ j$

Where, $i, j, k \in S,\ \ \forall i, j, k$

$SS_{ij} - trasited\ state\ sequence.$ *It is fixed for all the correct state transition from state i to state j.*

$SS_{ij}$ for all intermediate state transitions in *Ūrdhva Hāstāsana* is as follows,

$SS_{ac} = (a)$, $SS_{cd} = (a, c)$, $SS_{de_1} = (a, c, d)$, $SS_{e_1e} = (a, c, d, e_1)$, $SS_{ed} = (a, c, d, e_1, e)$, $SS_{ed} = (a, c, d, e_1, e)$, $SS_{dc} = (a, c, d, e_1, e, d)$, $SS_{ca} = (a, c, d, e_1, e, d, c)$

$t_k - frequency\ count\ for\ gesture\ k$

$\delta_e$ is error that has values for angle and height other than correct possible states and $e \notin S_{UH}$.

Correct expected sequence of postures in *Ūrdhva Hāstāsana* is,

aaa...aaaccc⋯cccddd⋯ddde$_1$e$_1$e$_1$⋯e$_1$e$_1$e$_1$eee⋯eeeddd⋯dddcccc⋯cccaaa⋯aaa

i.e. acde$_1$edca

Above discussed model is capable of assessing correctness of performed *Yogāsana* but can't comment on suggestion of amendment for incorrect postures. In order to suggest amendment, possible errors needs to be analyzed along with its position. Model for correctness assessment of *Ūrdhva Hastāsana* is modified as shown in figure 3.24. States indicated with *s'* are erroneous states. It is assumed that practitioner tried to perform posture *s* but done it erroneously, where *s* is any correct state present in *Yogāsana* $s \in S_{UH}$.

Possible errors while performing *Yogāsana* are:

**Skipped postures in *Āsana* sequence:** While performing *Āsana* practitioner may skip one or more postures. State transition diagram of *Ūrdhva Hāstāsana* shows such transitions with brown color. Posture sequence such as cde$_1$edca (Samasthiti skipped in pre-postures), ae$_1$edca (Hands apart and hands up skipped in pre-postures), acde'eda (Hands apart skipped in post-posture sequence), acdca (Skipped main posture) are examples of sequences where one or more postures are skipped.

**Erroneous postures:** While performing *Āsana* practitioner may perform one or more postures erroneously. State transition diagram of *Ūrdhva Hāstāsana* shows such transitions with brown color. Posture sequence such as a'cde$_1$edca (*Samasthiti* performed erroneously in pre-posture sequence), ac'd'e$_1$edca (Hands apart and hands up i.e. c' and d' respectively are done erroneously in pre-posture sequence), acde$_1$edc'a (Hands apart done erroneously in post-posture sequence), acde'dca (main posture performed erroneously) are examples of sequences where one or more postures are done erroneously.

**Erroneous End:** Practitioner may stop performing *Āsana* at intermediate position. This is erroneous end and is shown with green color in state transition diagram of *Ūrdhva Hāstāsana in* figure 3.24. Posture sequence such as ac (Erroneous end after two correct pre-postures), ac'd'e$_1$ (Erroneous end after e$_1$, practitioner performed correct *Samasthiti*, erroneous postures c' and d' and correct posture e$_1$), a'c' (Erroneous end after two incorrect postures) are examples of sequences where end erroneously at intermediate states.

Amendment is suggested at two levels: (I) Abstract level amendment in terms of missing / erroneous posture of erroneous end verses correct expected sequence, and (II) Detailed amendment in terms of body angle for missing / erroneous postures.

Mathematical model designed for assessment of correctness needs to be modified in order to suggest amendment. To identify the position of occurrence of error new term $sfn_i$ is introduced. $sfn_i$ will maintain starting position of corresponding posture (in terms of frame number) of each state.

$$AY_{UH} = \{(a, sfn_a, n_a), (c, sfn_c, n_c), (d, sfn_d, n_d), (\mathrm{e}', sfn_{e'} n_{e'}), (\mathrm{d}, sfn_{dd}, n_{dd}), (\mathrm{c}, sfn_{cc}, n_{cc}), (\mathrm{a}, sfn_{aa}, n_{aa})\}$$

$AS_{UH}$ represents all distinct states present in *Ūrdhva Hāstāsana* including all possible error states and is given by,

$$AS_{UH} = \{a, c, d, e_1, e, a', c', d', e', End_{error}, End\}$$

Transitions $SS_{ij'}$, $SS_{i'j}$ and $SS_{i'j'}$ are transitions from correct posture to incorrect posture, incorrect posture to correct posture and incorrect posture to incorrect posture respectively,$\forall ij \in S$.

Correct expected transition is acde$_1$edca shown with blue color. All other are incorrect transitions. Red color shows missing or erroneous intermediate poses, whereas green

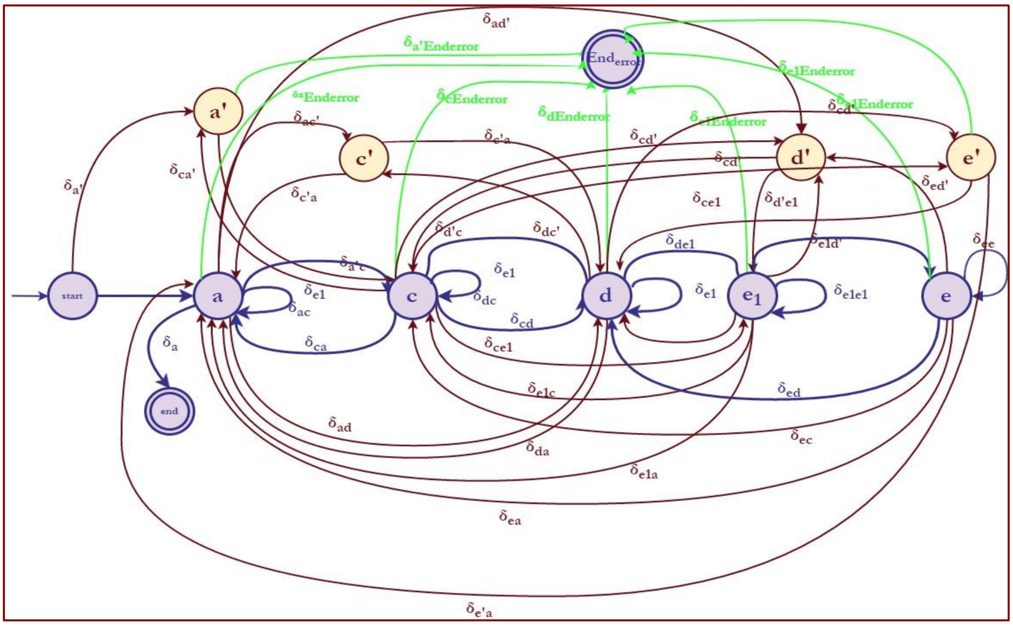



*Figure 3.24: State transition diagram for recognition of Ūrdhva Hāstāsana*

color shows erroneous termination of *Yogāsana*. Remaining mathematical model is used as it is from recognition of *Ūrdhva Hastāsana*.

Similar state transitions are prepared and used for correctness assessment of other mentioned *Yogāsana* and discussed in brief.

### *3.3.2. Samasthiti Tādāsana*

Figure 3.25 (A) and (B) shows state transition representation of *Samasthiti Tādāsana*. Identified distinct postures in *Samasthiti Tādāsana* are $S_{ST} = \{a, \mathrm{b}, error\}$. Figure 3.25 (C) shows state transition for assessment of correctness of *Āsana* and it can be represented as $Y_{ST} = (\langle a, n_a\rangle, \langle b_1, 290\rangle, \langle b, n_b\rangle, \langle a, n_a\rangle)$.

Amendment in *Āsana* is $AY_{ST} = (\langle a, n_a, sfn_a\rangle, \langle b_1, 290, sfn_{b1}\rangle, \langle b, n_e, sfn_a\rangle, \langle a, n_a, sfn_a\rangle)$

$AS_{ST}$ represents all distinct states present in *Samasthiti Tādāsana* including possible error states and is given by,

$$AS_{ST} = \{a, b, a', b', End_{error}\}$$

Transitions $SS_{ij'}$, $SS_{i'j}$ and $SS_{i'j'}$ are transitions from correct posture to incorrect posture, incorrect posture to correct posture and incorrect posture to incorrect posture respectively. $\forall i, j \in S_{ST}$.

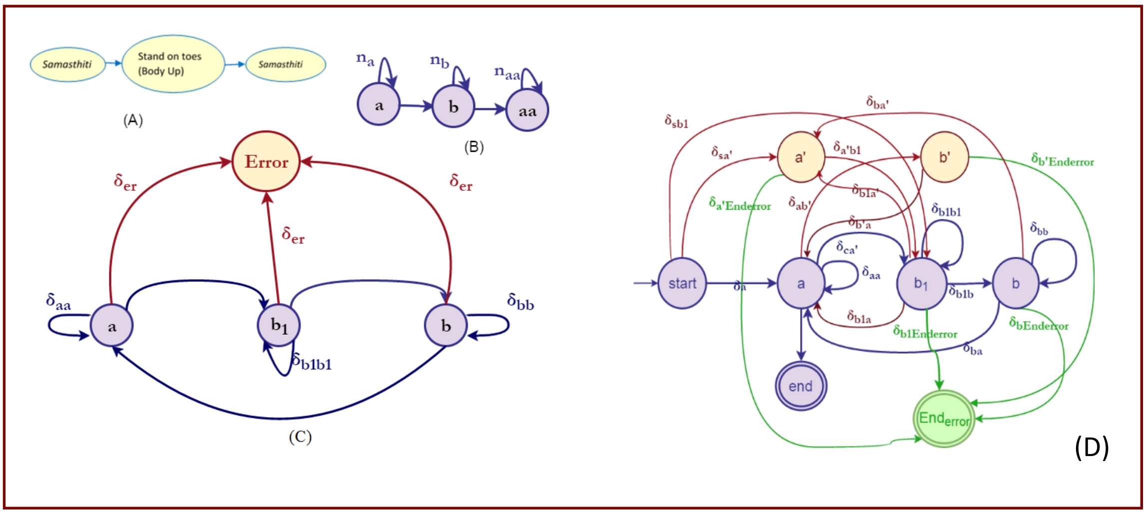



*Figure 3.25: State transition diagram for Samasthiti Tādāsana (A) Simple model, (B) Symbolic representation, (C) Recognition model, (D) Model for suggestion of amendment*

### *3.3.3. Ūrdhva Baddhanguliyāsana*

Figure 3.26 (A) and (B) shows state transition representation of *Ūrdhva Baddhanguliyāsana*. Identified distinct postures in *Ūrdhva Baddhanguliyāsana* are

$S_{UB} = \{a, \mathrm{c}, f, q, error\}$. Figure 3.26 (C) shows state transition for assessment of correctness of *Āsana* and it can be represented as,

$$Y_{UB} = \{\langle a, n_a\rangle, \langle c, n_c\rangle, \langle f, n_f\rangle, \langle q_1, n_{q_1}\rangle, \langle q, n_q\rangle, \langle f, n_f\rangle, \langle c, n_c\rangle, \langle a, n_a\rangle\}$$

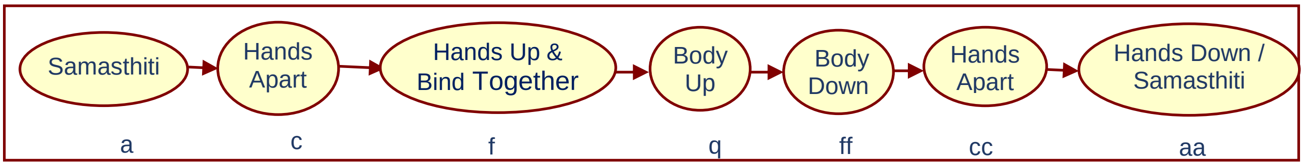
State transition sequence for amendment in *Ūrdhva Baddhanguliyāsana* is,

$$Y_{UB} = \{(a, sfn_a, n_a), (c, sfn_c, n_c), (f, sfn_f, n_f), (q_1, sfn_{q_1} n_{q_1}), (\mathrm{q}, sfn_q, n_q), (f, sfn_f, n_f), (c, sfn_c, n_c), (\mathrm{a}, sfn_a, n_a)\}$$

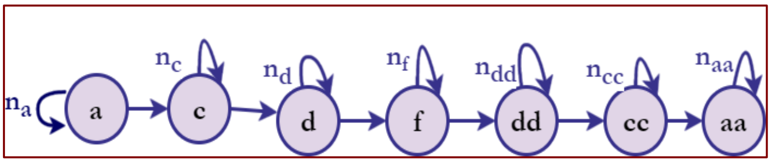
$$AS_{UB} = \{a, c, f, q_1, q, a', c', f', q', End_{error}, End\}$$

$AS_{UB}$ represents all distinct states present in *Ūrdhva Baddhanguliyāsana* including all possible error states.

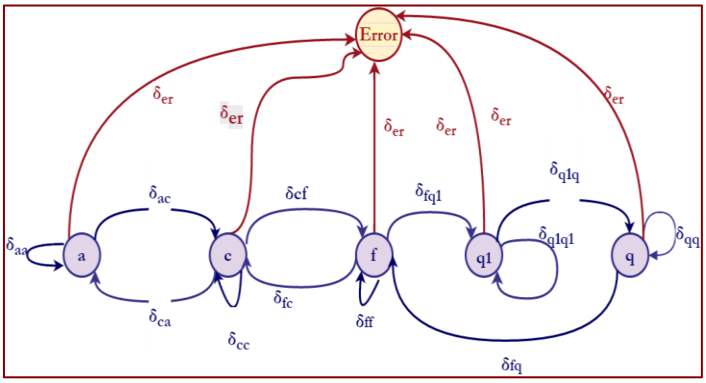
Transitions $SS_{ij'}$, $SS_{i'j}$ and $SS_{s's'}$ are transitions from correct posture to incorrect posture, incorrect posture to correct posture and incorrect posture to incorrect posture respectively. $\forall i, j \in S_{UB}$



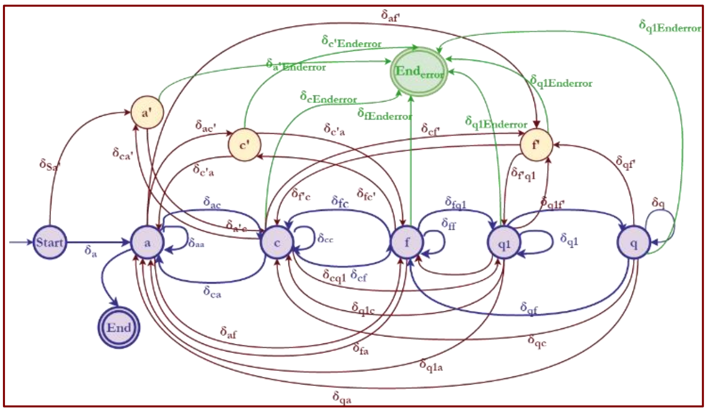

*(A)*




*(B)*



*(C)*



(D)

*Figure 3.26: State transition diagrams for recognition Ūrdhva Baddhanguliyāsana (A) Simple model, (B) Symbolic representation, (C) Recognition model, (D) Model for suggestion of amendment*

### 3.3.4. *Vīrabhadrāsana-II* (Warrior Pose-II)

Figure 3.27 (A) and (B) shows state transition representation of *Vīrabhadrāsana-II*. Identified distinct postures in *Vīrabhadrāsana-II* are $S_{VB} = \{a, \mathrm{g}, h, lr, ll, error\}$. Figure 3.27 (C) shows state transition for assessment of correctness of *Vīrabhadrāsana-II* and it can be represented as,

$$Y_{VBR} = \left(\langle a, n_a\rangle, \langle g, n_g\rangle, \langle h, n_h\rangle, \langle lr_1, 290\rangle, \langle lr, n_{lr}\rangle, \langle h, n_h\rangle, \langle g, n_g\rangle,\ \langle a, n_a\rangle\right)$$

$$Y_{VBL} = \left(\langle a, n_a\rangle, \langle g, n_g\rangle, \langle h, n_h\rangle, \langle ll_1, 290\rangle, \langle ll, n_{ll}\rangle, \langle h, n_h\rangle, \langle g, n_g\rangle,\ \langle a, n_a\rangle\right)$$

Amendment in *Vīrabhadrāsana-II is*

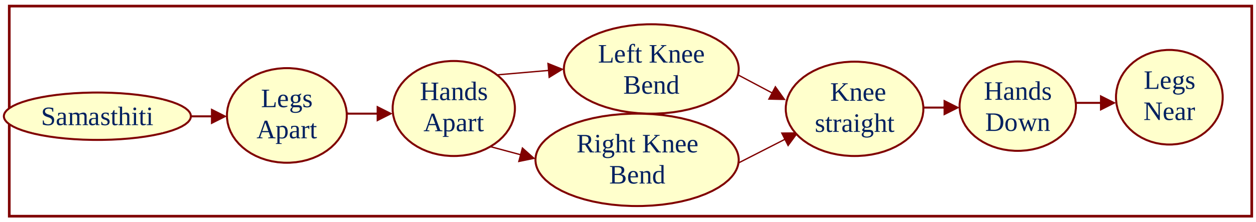
$$Y_{VBR} = \left(\langle a, sfn_a, n_a\rangle, \langle g, sfn_g, n_g\rangle, \langle h, sfn_h, n_h\rangle, \langle lr_1, sfn_{ll1}, 290\rangle, \langle lr, sfn_{ll}, n_{lr}\rangle, \langle h, sfn_h, n_h\rangle, \langle g, sfn_g, n_g\rangle,\ \langle a, sfn_a, n_a\rangle\right)$$

$Y_{VBL}$
$= (\langle a, sfn_a, n_a\rangle, \langle g, sfn_g, n_g\rangle, \langle h, sfn_h, n_h\rangle, \langle lr_1, sfn_{lr1}, 290\rangle, \langle lr, sfn_{lr}, n_{lr}\rangle, \langle h, sfn_h, n_h\rangle, \langle g, sfn_g, n_g\rangle, \langle a, sfn_a, n_a\rangle)$

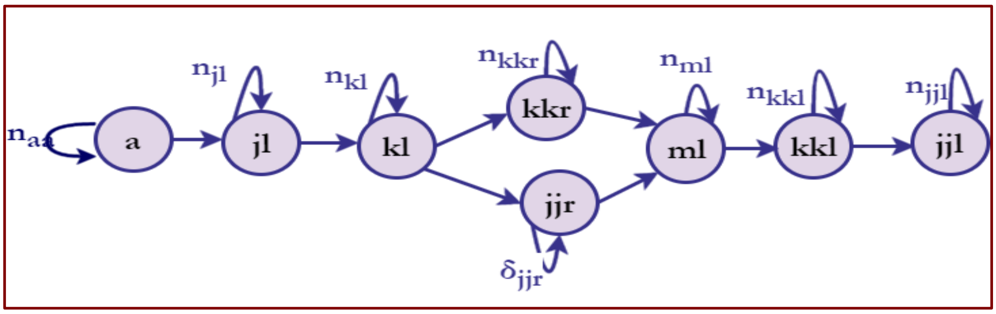
$$AS_{VBR} = \{a, g, h, lr_1, \mathrm{lr}, a', c', f', q', End_{error}\}$$

$AS_{VBR}$ represents all distinct states present in *Vīrabhadrāsana-II* including all possible error states.

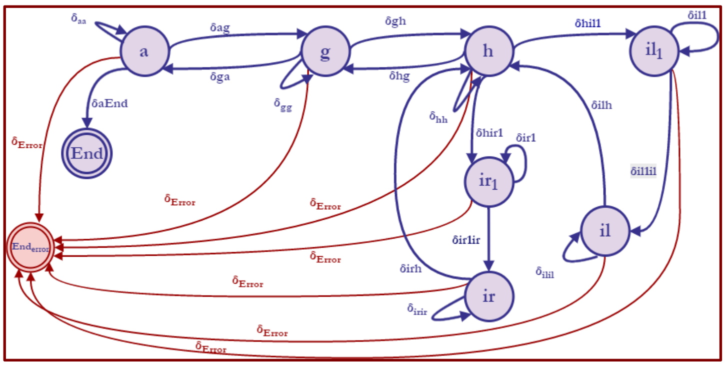
Transitions $SS_{ij'}$, $SS_{i'j}$ and $SS_{i'j'}$ are transitions from correct posture to incorrect posture, incorrect posture to correct posture and incorrect posture to incorrect posture respectively. $\forall i, j \in S_{VB}$



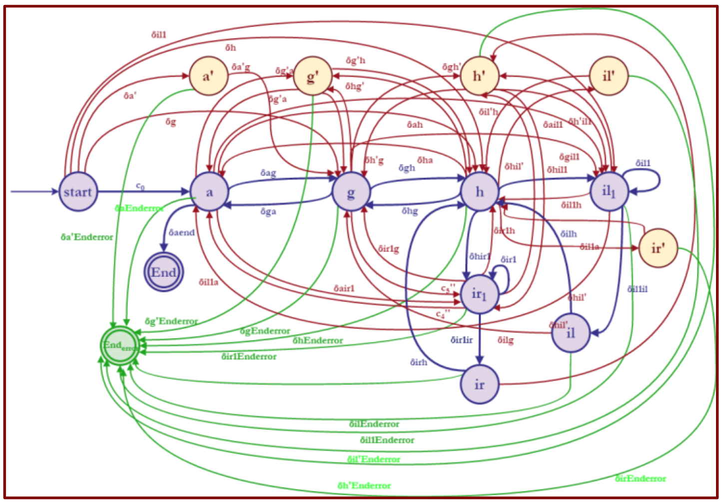

*(A)*




*(B)*




*(C)*




*(D)*

*Figure 3.27: State transition diagram for Vīrabhadrāsana-II (A) Simple model, (B) Symbolic representation,(C)Recognition model, (D) Model for suggestion of amendment*

### ***3.3.5. Vṛukṣāsana***

Figure 3.28 (A) and (B) shows state transition representation of *Vṛukṣāsana*. Identified distinct postures in *Vṛukṣāsana* are $S_V = \{a, \mathrm{jr}, \mathrm{kr}, mr_1, mr, jl, kl, ml_1, ml, End_{error}, End\}$. Figure 3.28 (C) shows state transition for assessment of correctness of *Vṛukṣāsana* and it can be represented as,

$$Y_{VR} = \left(\langle a, n_a\rangle, \langle jr, n_{jr}\rangle, \langle kr, n_{kr}\rangle, \langle mr_1, 290\rangle, \langle mr, n_{mr}\rangle, \langle kr, n_{kr}\rangle, \langle jr, n_{jr}\rangle, \langle a, n_a\rangle\right)$$

$$Y_{VL} = \left(\langle a, n_a\rangle, \langle jl, n_{jl}\rangle, \langle kl, n_{kl}\rangle, \langle ml_1, 290\rangle, \langle ml, n_{ml}\rangle, \langle kl, n_{kl}\rangle, \langle jl, n_{jl}\rangle, \langle a, n_a\rangle\right)$$

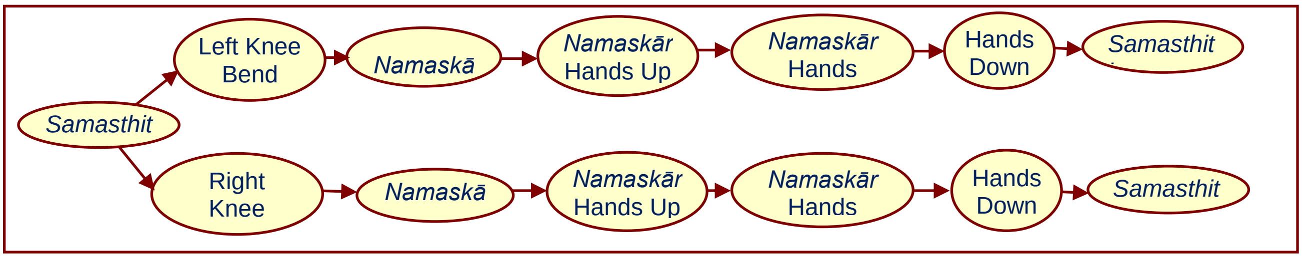
Amendment in *Vṛukṣāsana* is,

$$Y_{VR} = \left(\langle a, sfn_a, n_a\rangle, \langle jr, sfn_{jr}, n_{jr}\rangle, \langle kr, sfn_{kr}, n_{kr}\rangle, \langle mr_1, sfn_{mr1}, 290\rangle, \langle mr, sfn_{mr}, n_{mr}\rangle, \langle kr, sfn_{kr}, n_{kr}\rangle, \langle jr, sfn_{jr}, n_{jr}\rangle, \langle a, sfn_a, n_a\rangle\right)$$

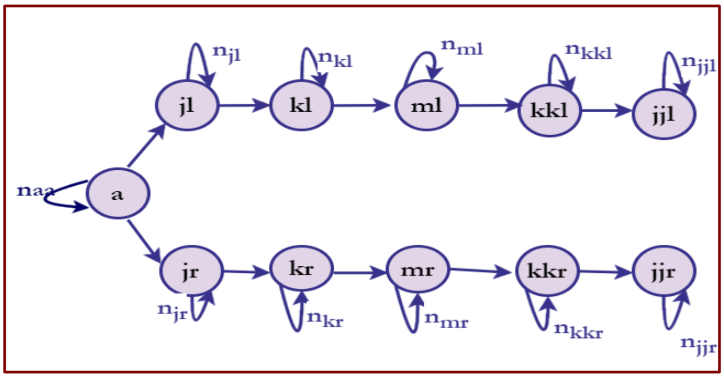
$$Y_{VL} = \left(\langle a, sfn_a, n_a\rangle, \langle jl, sfn_{jl}, n_{jl}\rangle, \langle kl, sfn_{kl}, n_{kl}\rangle, \langle ml_1, sfn_{ml1}, 290\rangle, \langle ml, sfn_{ml}, n_{ml}\rangle, \langle kl, sfn_{kl}, n_{kl}\rangle, \langle jl, sfn_{jl}, n_{jl}\rangle, \langle a, sfn_a, n_a\rangle\right)$$

$$AS_{VR} = \{a, g, h, lr_1, \mathrm{lr}, a', c', f', q', End_{error}\}$$

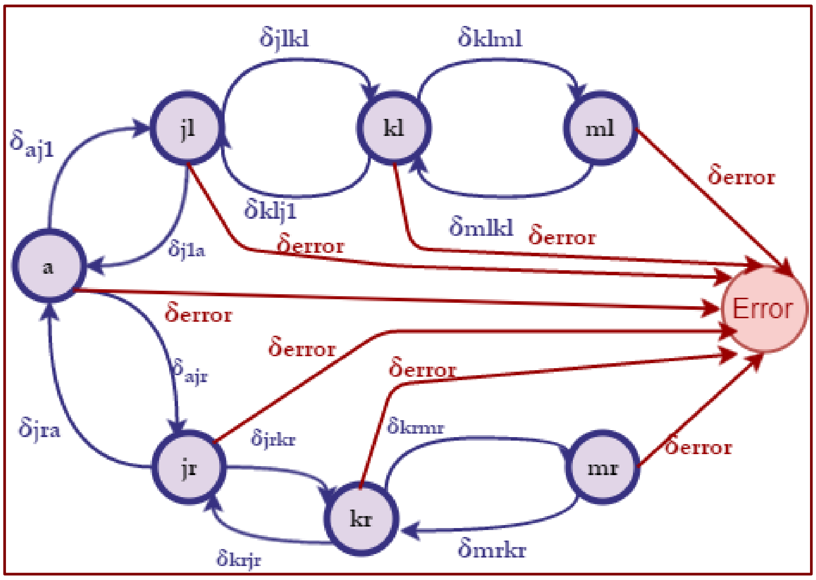
$AS_{VR}$ represents all distinct states present in *Vṛukṣāsana* including all possible error states. Transitions $\delta_{ij'}$, $\delta_{i'j}$ and $\delta_{i'j'}$ are transitions from correct posture to incorrect posture, incorrect posture to correct posture and incorrect posture to incorrect posture respectively. $\forall i, j \in S_{VR}$.


Samasthit
Left Knee Bend
Namaskā
Namaskār Hands Up
Namaskār Hands
Hands Down
Samasthit
Right Knee
Namaskā
Namaskār Hands Up
Namaskār Hands
Hands Down
Samasthit

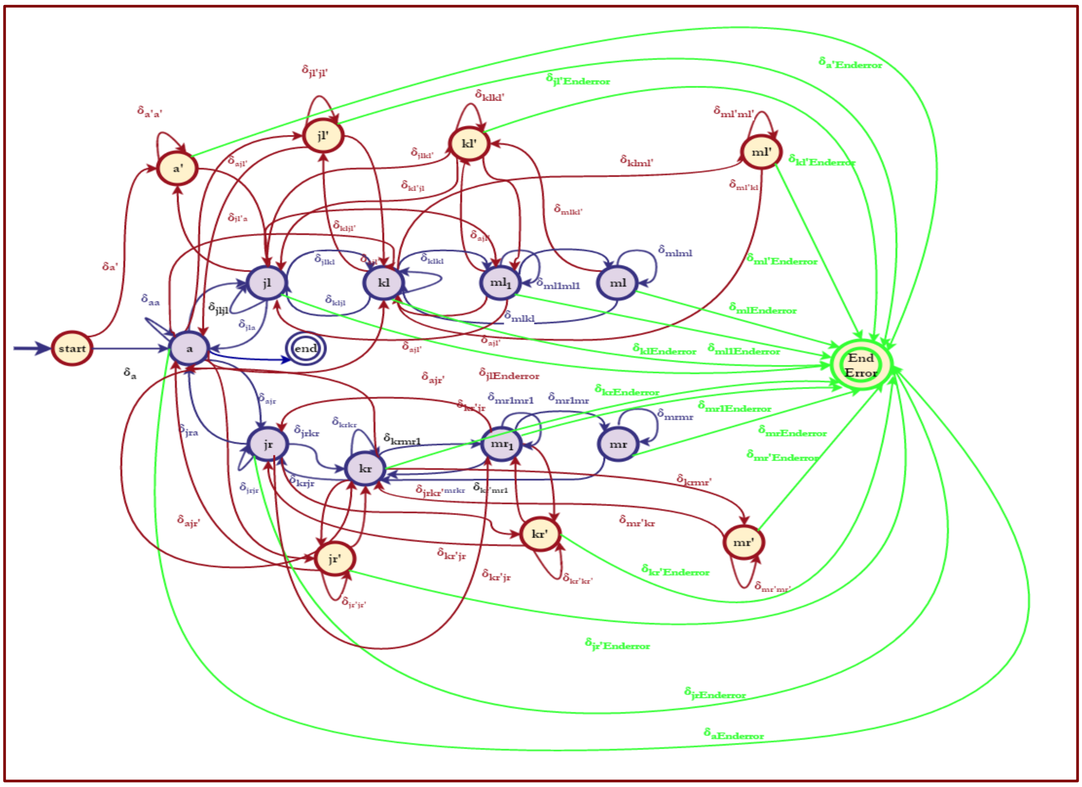
*(A)*


naa
a
jl
kl
ml
kkl
jjl
njl
nkl
nml
nkkl
njjl
jr
kr
mr
kkr
jjr
njr
nkr
nmr
nkkr
njjr

*(B)*


δjlkl
δklml
δaj1
jl
kl
ml
δerror
δklj1
δmlkl
δj1a
δerror
δerror
a
Error
δerror
δerror
δajr
δerror
δjra
δjrkr
δkrmr
jr
mr
δerror
kr
δkrjr
δmrkr

*(C )*




(D)

*Figure 3.28: State transition diagram for recognition of Vṛukṣāsana (A) Simple model, (B) Symbolic representation, (C) Recognition model, (D) Model for suggestion of amendment*

### *3.3.6. Ardhauttānāsana*

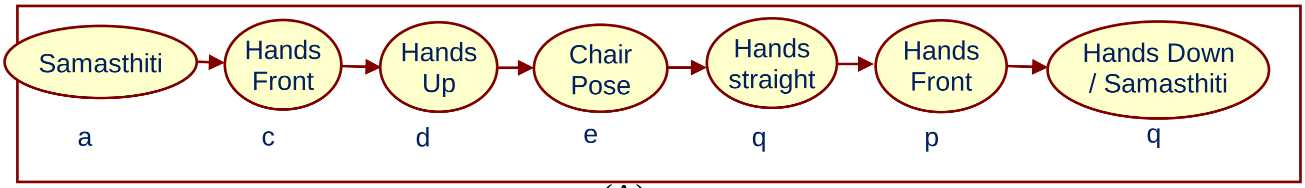
Figure 3.29 (A) and (B) shows state transition representation of *Ardhauttānāsana*. Identified distinct postures in *Ardhauttānāsana* are $S_{AU} = \{a_s, n_s, d_s, o_{s1}, o_s,\ error\}$. Figure 3.29 (C) shows state transition for assessment of correctness of *Āsana* and it can be represented as,

$$Y_{AU} = \big(\langle a_s, n_s\rangle, \langle n_s, n_{ns}\rangle, \langle d_s, n_{ds}\rangle, \langle p_{s1}, 290\rangle, \langle p_s, n_{ml}\rangle, \langle d_s, n_{ds}\rangle, \langle n_s, n_{ns}\rangle,\ \langle a_s, n_s\rangle\big)$$

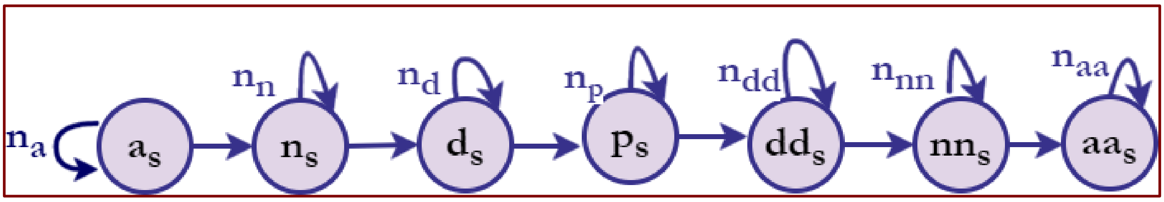
$$Y_{AU} = \big(\langle a, sfn_s, n_s\rangle, \langle n_s, sfn_a, n_{ns}\rangle, \langle d_s, sfn_a, n_{ds}\rangle, \langle p_{s1}, sfn_a, 290\rangle, \langle p_s, sfn_a, n_{ml}\rangle, \langle d_s, sfn_a, n_{ds}\rangle, \langle n_s, sfn_s, n_{ns}\rangle,\ \langle a, sfn_s, n_s\rangle\big)$$

$$AS_{AU} = \{a_s, n_s, d_s, o_{s1}, o_s, a_s', n_s', d_s', o_{s1}', o_s', End_{error}\}$$

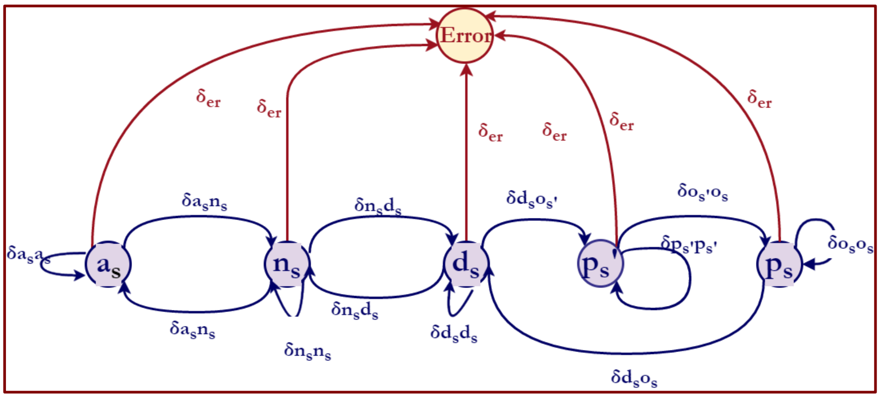
$AS_{AU}$ represents all distinct states present in *Ardhauttānāsana* including all possible error



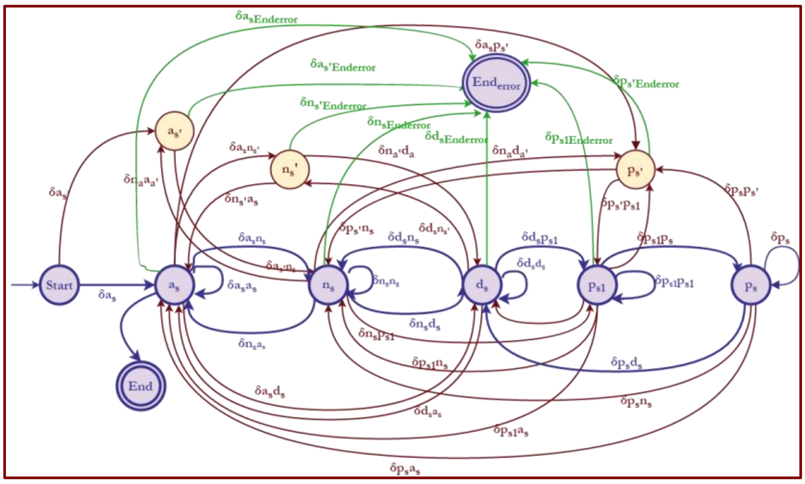

(A)




(B)




(C)



*(D)*

*Figure 3.29: State transition diagram for recognition of Ardhauttānāsana (A) Simple model, (B) Symbolic representation, (C) Recognition model, (D) Model for suggestion of amendment*

### 3.3.7. *Utkaṭāsana*

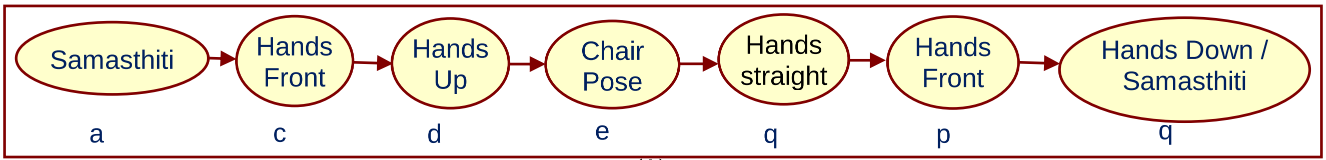
Figure 3.30 (A) and (B) shows state transition representation of *Utkaṭāsana*. Identified distinct postures in *Utkaṭāsana* are $S_U = \{a_a, n_a, d_a, o_{a1}, o_a,\ error\}$. Figure 3.30 (C) shows state transition for assessment of correctness of *Utkaṭāsana* and it can be represented as,

$$Y_U = \left(\langle a_a, n_{aa}\rangle, \langle n_a, n_{na}\rangle, \langle d_a, n_{da}\rangle, \langle o_{a1}, 290\rangle, \langle o_a, n_{oa}\rangle, \langle d_a, n_{da}\rangle, \langle n_a, n_{na}\rangle,\ \langle a, n_a\rangle\right)$$

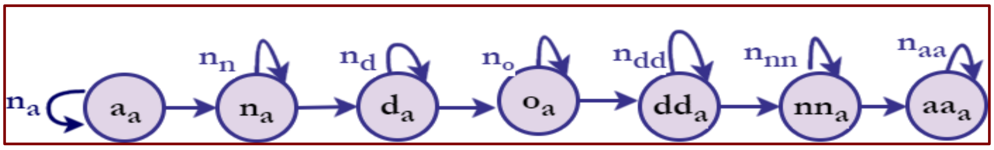
$$Y_U = \left(\langle a_a, sfn_a, n_{aa}\rangle, \langle n_a, sfn_a, n_{na}\rangle, \langle d_a, sfn_a, n_{da}\rangle, \langle o_{a1}, sfn_a, 290\rangle, \langle o_a, sfn_a, n_{oa}\rangle, \langle d_a, sfn_a, n_{da}\rangle, \langle n_a, sfn_a, n_{na}\rangle,\ \langle a, sfn_a, n_a\rangle\right)$$

$$AS_U = \{a_a, n_a, d_a, o_{a1}, o_a, a_a{}', n_a{}', d_a{}', o_{a1}{}', o_a{}', End_{error}\}$$

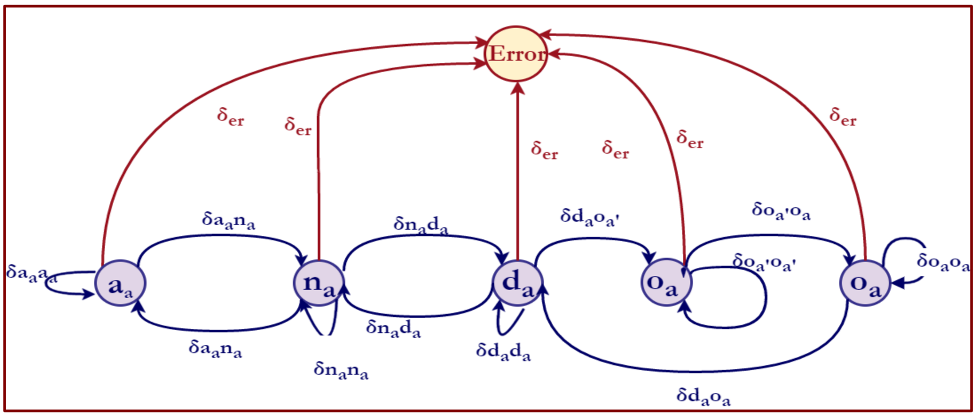
$AS_U$ represents all distinct states present in *Utkaṭāsana* including all possible error states.



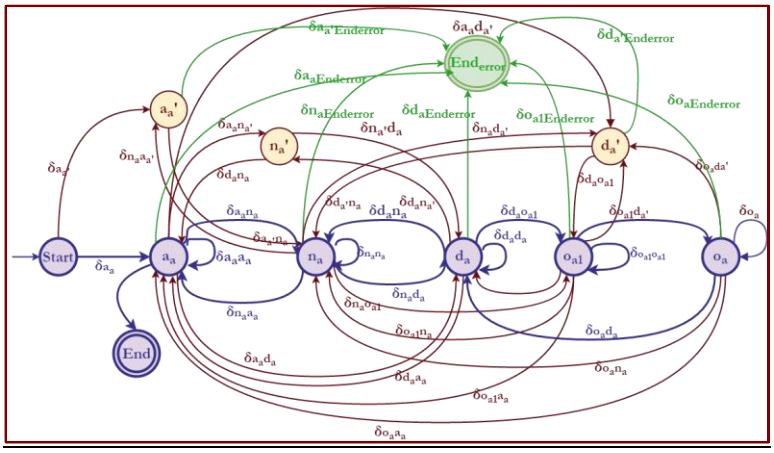

*(A)*




*(B)*




*(C)*




*(D)*

*Figure 3.30: State transition diagram for recognition of Ārdha Uttānāsana (A) Simple model, (B) Symbolic representation,(C)Recognition model, (D) Model for suggestion of amendment*

## 3.4. Algorithms

Three algorithms are designed two for recognition of *Yogāsana* and one for suggestion of amendment in *Yogāsana*. Algorithm in section 3.3.1 gives rule based system for recognition of *Yogāsana* on real time data and human joint features are used for recognition. Algorithms in 3.3.2 recognizes *Yogāsana* using angle features and two layer hierarchical model using state transitions. Algorithms in 3.3.3 suggest amendment in *Yogāsana* using angle features and three layer hierarchical model using state transitions for amendment.

### 3.4.1. Real Time *Yogāsana* Recognition System using Joint Feature

*Yogāsana* is transition of posture sequence from one state to another.

$$Yogasana = \{P_{1,}\ P_{2,}\ P_{3,}\ \dots\dots,\ P_n\}$$

Where, $P_i$ is posture in given *Āsana* $1 \le i \le n$, n

Each posture is given by articulated human model represented using 20 joints

Each joint is represented using *(X, Y)* values

Posture $\quad P_i = \{\{J^1_{i,x},\ J^1_{i,y}\},\ \dots,\ \{\{J^{20}_{i,x},\ J^{20}_{i,y}\}\}$

Samasthiti is taken as reference posture for initialization

$$P_1 = \ \left\{\{J^1_{1,x},\ J^1_{1,y}\},\cdots,\ \{J^{20}_{1,x},\ J^{20}_{1,y}\}\right\}$$

Posture in sequence is identified from difference of most informative joints of reference posture and recent posture. *Diff* $(P_i, P_1)$

$$\{\delta_{i,x}^{k},\ \delta_{i,y}^{k}\} = diff\left(\{J_{1,x}^{k},\ J_{1,y}^{k}\},\{J_{i,x}^{k},\ J_{i,y}^{k}\}\right), \qquad 1 \le i \le 10$$

Rules are defined by setting value of $\{\delta_{i,x}^{k},\ \delta_{i,y}^{k}\}$ for each posture and each posture is computed by comparing the values. *Yogāsana* is recognized using simple state model. Algorithm is tested on three *Yogāsana* - *Samasthiti Tādāsana*, *Ūrdhva Hastāsana, Vīrabhadrāsana-II.*

### 3.4.2. Algorithms for Two Layer Hierarchical Model for Recognition of *Yogāsana*

Algorithm 1 gives procedure for recognition of *Yogāsana* from skeleton data. Human joint positions are provided as input and algorithm gives *Yogāsana* name as output. Detailed procedure for knowledge-base preparation on expert's postures as discussed in section 3.2.1 is given in Algorithm 2. Designing knowledge-base is one time process and knowledge-base once created can be used for recognition of all included *Yogāsana* of all practitioner. Knowledge-base is created for all gestures shown in Table 1 using Algorithm 2 and stored in $[YKB]_{pX11}$. For recognition of different actions other than *Yogāsana* key-postures need to be identified and by changing knowledge-base with identified postures, system will become capable of recognizing new action set.

To create knowledge-base first body angles need to be computed for all the postures as given in step 2 in Algorithm 2. Computation of range of angles from experts' data and allowed deviation suggested by trainer for each body angle using anthropometric parameters is computed as explained in figure 3.18. Mathematically it can be represented as shown by equation (2) and equation (3). Equation (2) computes mode value of all the frames for each expert. Equation (3) gives min and max value for each body angle. Obtained min max range is further modified in consultation with trainer and stored in knowledge-base.

**Algorithm 1:** Recognition of *Yogāsana* from Skeleton Data

**Input:** Kinect Sensor Data / Human joint values in the form of (X, Y) co-ordinate

**Output:** Name of Recognized *Yogāsana*

**1:** ***Procedure*** *RecognizeYogāsana ()*

**2:** *[YKB]*$_{pX11}$ = knowledge-base using experts' data for each gesture in Table 3.2 using Algorithm 2.

**3:** *Yogāsana* = Recognized *Yogāsana* using Algorithm 4.

**4:** ***end Procedure***

$$A^{(j)}_{i\,mode} = \underset{1\le k\le n}{\text{mode}}\left\{A^{(j)}_{ki}\right\} \qquad (2)$$

$$a_{i\,min} = \min_{1\le j\le m}\left\{A^{(j)}_{i\,mode}\right\} \text{ and } a_{i\,max} = \max_{1\le j\le m}\left\{A^{(j)}_{i\,mode}\right\} \qquad (3)$$

*i – Body angles* $1 \le i \le 10$

*n – Number of frames for each expert*

*m – Number of experts considered for creation of knowledge base*

Body angles are computed from human joint positions using Algorithm 3. Algorithm 3 uses 360$^{o}$ geometry for angle computation as explained in equation (1).

Algorithm 4 gives procedure for recognition of *Yogāsana*. Input to algorithm is practitioner's skeleton data, knowledge-base of angle features prepared on expert's data *[YKB]*$_{pX11}$ computed using Algorithm 2 and expected correct posture sequence in *Yogāsana*. Angle features are computed using algorithm 3. Number of angles may vary with type of *Yogāsana* and angles may also change with change in view point. First, each *Yogāsana* frame is labeled with most matched posture from knowledge-base. Euclidean Distance (ED) is used for finding closeness of each frame and identified postures.

**Algorithm 2:** Develop Knowledge-Base for Recognition

**Input:** Captured posture data for each posture states in the form of (x, y) stream for 20 joints $[JV]_{m\ X\ 40}$ where m is number of frames.

**Output:** Knowledge Base on Expert's data $[YKB]_{\ p\ X\ 11}$

1: ***Procedure*** *YogāsanaKnowledgeBase*()

*2:* **for** $i\ =\ 1\ \boldsymbol{to}\ NoofPosture$

2.1. Compute angles between Expert's body parts for representation of posture using *Algorithm 3* for angles shown in figure 3.11.

2.2. [EGV] $_{i\ X\ 11}$ *MODE* value for all angles in posture

3: **End for**

4: Design knowledge base $[YKB]_{\ p\ X\ 11}$ by storing Min and Max values for each angle using expert posture's computed body angles and allowed deviation *δ1, δ2,…, δ10* suggested by trainer in angle at joint position.

5: **Return** $[YKB]_{p\ X\ 11}$

*6:* ***end procedure***

**Algorithm 3:** Angle computation from given three points out of $360^0$

**Input:** Three points *A* $(X_1, Y_1)$, $B(X_2, Y_2)$, $C(X_3, Y_3)$

**Output:** Angle ∡B stored in ang

**1: procedure** *Angle360(A,B,C)*

// Gives clockwise angle between two vectors

**2:** $dot = X_1\text{-}X_2{*}X_3\text{-}X_2{+}Y_1\text{-}Y_2{*}Y_3\text{-}Y_2$

**3:** $det = X_1\text{-}X_2{*}Y_3\text{-}Y_2\text{-}Y_1\text{-}Y_2{*}X_3\text{-}X_2$

**4:** *ang = atan2(det, dot)*

**5:** $ang1 = ang * 180 * 7/22$

**6:** $\boldsymbol{if}(ang1 < 0)\ \boldsymbol{then}$

**7:** $\quad ang1 = -ang$

**8:** $\boldsymbol{else}$

**9:** $\quad ang = 360 - ang$

**10:** ***endif***

**11:** ***end procedure***

**Algorithm 4** *Yogāsana* Recognition

**Input:** Skeleton data of practitioner$[JV]_{f X jtpt}$, Expert's Knowledge base $[YKB]_{pXangle}$

*jtpt – body joint points X 2, angle – total angles considered for recognition*

**Output:** Name of Recognized *Yogāsana*

**1:** ***Procedure*** *YogāsanaRecognition()*

**2:** Compute angle feature for each frame of practitioner's posture data using Algorithm 2 for angles shown in figure 3.11.

**3:** *[PD]* $_{frame\ X\ (angle+1)}$ = computed angles along with frame number.

**4:** Compute Euclidean Distance *ED* between body angle feature of individual posture in each practitioner's *Āsana* frame and each posture from knowledge base as shown in figure 3.19.

$$[post]_{frameX\ (p+1)} = ED_{p}^{f}{}_{1 \le f \le frame, 1 \le p \le posture}$$

$$= \sqrt{\sum_{i=0}^{9} (Angle_i^f - Posture_i^p)^2}$$

**5:** Most matched postures $recg^{(f)} = \min_{1 \le d \le posture} \{ED_d^f\}$ i.e. with smallest ED array.

**6:** $[posture\ ]_{k\ X\ 2}$ = frequency count for most matched postures $(post\ no, freq\ count)$

**7:** Remove noise

**8:** $[postureFinal\ ]_{k\ X\ 2}$ = Add similar posture counts.

**9:** Verify *Āsana* sequence from knowledge base with obtained posture sequence using designed state model.

**10:** *Āsana* = Recognized *Āsana* Name

***11: end procedure***

$$ED = \sqrt{\sum_{i=0}^{9} (A_i - P_i)^2}$$

Most matched posture is computed using step 5 and 6. $recg^{(f)} = \min_{1 \le d \le posture} \{ED_d^f\}$, $f$ is frame number. Details are shown in figure 3.19. Frequency count of most matched postures is maintained in $[posture]_{k\,X\,2}$, first column stores identified postures and second maintain count of occurrences. Very few intermediate frames where transition of movement takes place will be labeled as unidentified frames and all unidentified frames

are removed as noise. Again similar postures at consecutive places needs to be added and maintained in $[postureFinal\,]_{k\,X\,2}$ s. Now, we have series of ($post\,no$, $freq\,count$). Designed state transition models for recognition of *Āsana* shown in figure 3.21 to 3.27 and labeled with (C) are used for recognition of *Yogāsana*.

### 3.4.3. Suggestion of amendment in *Yogāsana*

Flowchart for suggestion of amendment in incorrectly performed *Yogāsana* is shown in figure 3.31. System first verifies posture sequence correctness. For correct posture sequence check for the main posture hold time. For novice practitioner if hold time is more than 10 seconds then performed *Yogāsana* is correct. If the posture hold time is less than required one then provide feedback about current hold time and required message for hold time.

For incorrect posture sequence, identify correct and erroneous / missing postures and give feedback to practitioner. Abstract level feedback provides name and position of missing / erroneous postures along with correct expected posture name. This is abstract level representation of amendment in performed incorrect *Yogāsana* that provides practitioner feedback about what is wrong? Feedback for, why it is incorrect is given at detailed amendment level. Detailed amendment is analyzed at three levels in pre-posture sequence, main-posture and post-posture sequence. It is represented in the form of frame numbers and details of incorrect body angle, along with correct expected range of angle. There can be single or multiple errors in *Yogāsana*.

First, the missing / erroneous posture in pre-posture sequence is analyzed and details are given. Similarly, the details of main posture and post posture are analyzed and displayed as feedback for suggestion of amendment in *Yogāsana*.

Algorithm 5 gives steps for suggestion of amendment in human action. For more accurate results simple moving average filter is used for data smoothing as shown in step 2 $\widehat{X_n} = \frac{1}{N+1}\sum_{i=0}^{N} X_{n-i}$ , $\forall\ frames$. Here, N is window size. *Yogāsana* knowledge base

**Algorithm 5:** Suggestion of Amendment in *Yogāsana*

**Input:** Kinect sensor skeleton data of practitioner$[JV]_{fXjtpt,}$ *jtpt – body joint points X 2, angle – total angles considered for recognition*, Name of *Yogāsana*, *[PD]* *frame X (angle+1)*

**Output:** Correctness Message, Details of Correction [Error] e X 10

1. ***procedure*** *AmendmentInYogāsana ()*
2. Apply Data Smoothing using Simple Moving Average Filter

   $$\widehat{X_n} = \frac{1}{N+1}\sum_{i=0}^{N} X_{n-i}\ , \forall frames$$
3. *[YKB]pX11* = knowledge-base using experts' data for each gesture in Table 3.2 using Algorithm 2.
4. $[Angle]_{f\ X\ angle+1}$ = Compute angle feature for each frame of practitioner's *Yogāsana* data using Algorithm 3 for all angles shown in figure 3.11.
5. *[PD]* *frame X (angle+1)* = computed angles along with frame number.
6. $[post\ ]_f\ =\ a_{f,\ \ i\,min,} \leq [Angle]_{fXi} \leq a_{f,\ \ i\,max}\ , \forall\ f, i$
7. $[posture\ ]_{k\ X\ 2}$ = frequency count for most matched posture from *[PD]* *frame X (angle+1)* in the form of $(post\ no, freq\ count)$
8. Remove noise
9. $[postureFinal\ ]_{k\ X\ 2}$ = Add similar posture counts.
10. Recognize Correctness of *Āsana* and suggestion of abstract level amendment using Algorithm 6

    $$Correctness = Y/N$$

    $$Message = Abstract\ level\ amendment$$
11. $[Error]_{e\ X\ 10}$ = Erroneous angles and correct expected angles using algorithm 7 i.e. detailed amendment in action
12. ***end procedure***

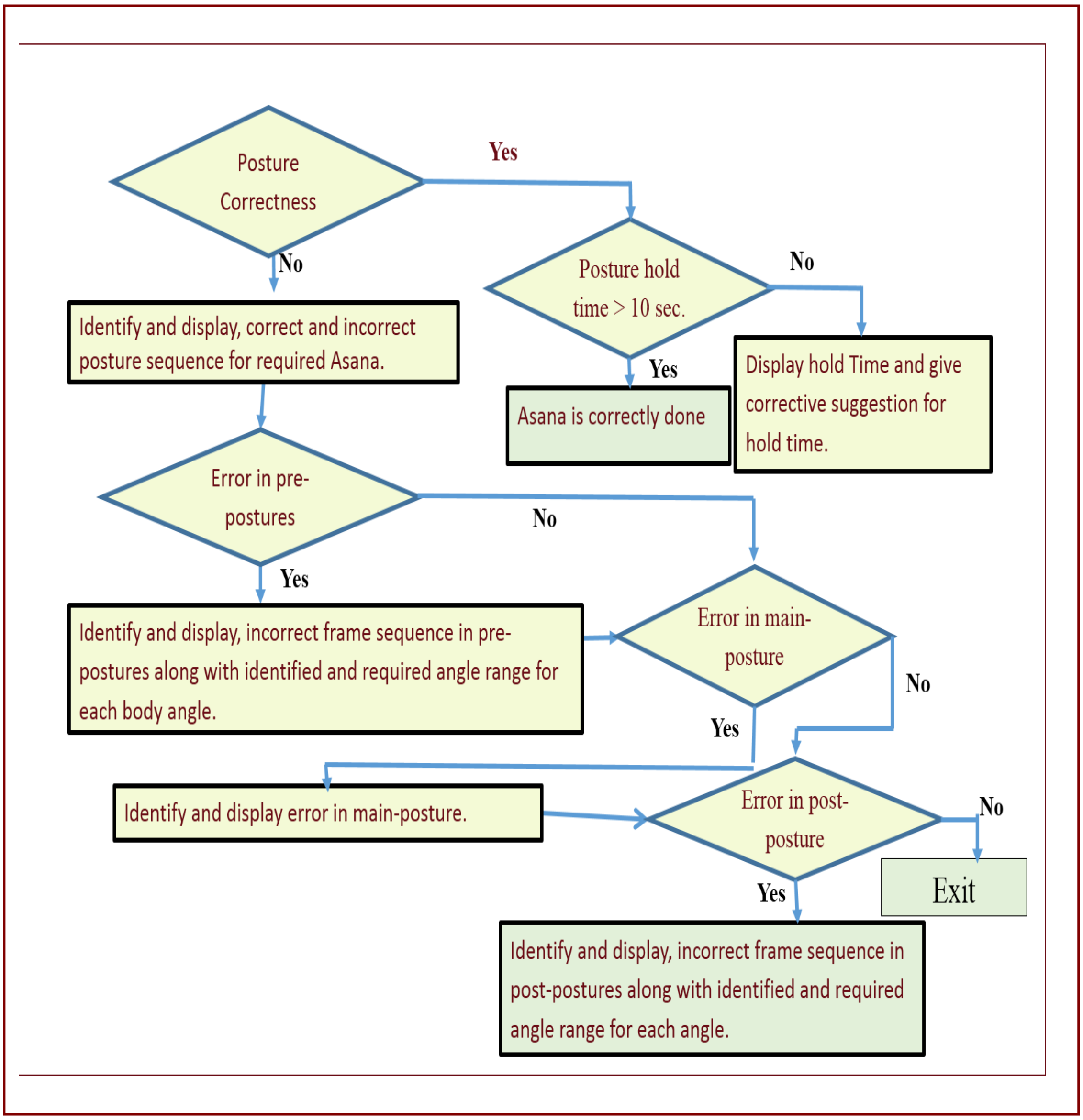



*Figure 3.31: Flowchart for suggestion of amendment*

*[YKB]$_{pX11}$* is computed using algorithm 2. Step 6 identifies the posture in frame by matching value with range of expert's data $a_{i\,min} \leq [Angle]_{fXi} \leq a_{i\,max}$. Step 7 to step 9 maintains posture name and posture count pair, that is given as input to algorithm 6 for abstract level assessment of *Yogāsana.* Correctness status of *Yogāsana* is maintained in *Correctness* and description in *Message*.

Algorithm 6 suggests abstract level amendment and step 2 to step 6 check for correct posture sequence. If the posture sequence is correct, check for hold time. It is done in step 7 to step 12. Step 13 to step 15 verifies for any extra movement. Step 16 to 37 provides feedback in terms of sequence of performed postures with correct and incorrect remark in *Message* variable.

**Algorithm 6 :** Assessment of Correctness

**INPUT:** Identified posture sequence along with hold time of main posture $[PostureFinal\ ]_{k\ X\ 2}$

k- number of postures identified and column 1 stores posture number and column 2 hold time of corresponding posture
M- Total number of postures in *Āsana.*
*[Asana]*$_{MX2}$ - *Āsana* postures

**OUTPUT:** Identifies Correctness of *Āsana*. If incorrect provides correction at abstract level appropriate correction message in *Message*.

***procedure*** *AssessĀsanaCorrectness()*
//Check for number of correct states in *Āsana*. Loop will break at un- matched location.
**for** $i\ =\ 0\ to\ k$ **do**
**if** $(postureFinal\ [i,0]!$
$= Asana[i,0])$ **then**
break
**end if**
**end for**
//For correct posture sequence check for hold time of main posture.
**if** $(\boldsymbol{i} = M)$ **then**
**if** $(postureFinal\ [m/2+1,0] < 300)$ **then**
Message = 'Current Hold Time is:' $GEST[3,1]/300)$. Increase hold time to 10 seconds or 3 breathings'
**else if** $(postureFinal\ [m/2+1,0] \geq 300)$ **then**
Message = '*Āsana* Performed correctly. You are doing well'
**end if**
**end if**
**if** $(k \geq i\ AND\ i = k)$ **then**
Message = 'Extra movements are done after *Āsana*'
**end if**
j = 0
//correct posture sequence
***while*** $j \leq k$ **do**
**if** $(postureFinal[j,0] = Asana[j,0])$ **then**
$CorrectPost[j] = Asana[j,0]$
**Else**
break;
**end if**
$\boldsymbol{j}++$
***end while***

*//incorrect posture sequence if any*

25. $\boldsymbol{while}(j \leq k)$ ***then***

26: **if** $(postureFinal[j,0] \neq Asana[j,0])$ **then**

27: $CorrectPost[j] = Asana[j,0]$

28: **Else**

29: break;

30: **end if**

*//Correct posture sequence if any*

31: $\boldsymbol{while}\ j \leq k$ **do**

32: **if** $(postureFinal[j,0] = Asana[j,0])$ **then**

33: $CorrectPost[j] = Asana[j,0]$

34: **end if**

35: $j++$

36: ***end while***

37: **return** Message

38: **end procedure**

Algorithm 7 provides detailed amendment in *Yogāsana.* Step 2 to step 15 identifies error in pre-posture sequence in terms of posture and frame sequence and correct required angle values and measured angle values. There is high possibility for novice practitioner of not holding main posture constantly and practitioner comes to previous posture and again try to attend main posture. It is shown in step 19 to step 28.

**Algorithm 7:** Detailed amendment in *Yogāsana*

**Input:** Detected *Āsana* sequence $[PostureFinal\,]_{k\ X\ 2}$

$[bodyAngle]_{frames\ X\ Angle}$- posture angles for practitioner data.

**Output:** Wrong angle list along with correct angle list

1: $g = gg = i$

//check error in pre-cycle

2: **if** $(i \leq n/2)$ **then**

3: Message = Error in Pre-postures

4: Identify frame number of incorrect posture.

5: Store starting and ending frame numbers for incorrect postures sequence in $startFrame$ and $endFrame$ respectively.

6: $diff = startFrame - endFrame$

7: if (*diff* = 0)

8: Message1 = *Āsana* posture not performed

9: else

10: Retrieve angle in frame which does not match with required posture angle

11: for $l = startFrame\ to\ endFrame$

12: for $n = 0\ to\ 10$

13: if $(PostureFinal[l, 7*n+m-1] = 0)$

14: IA = $W2[l, 7*n+m-1]$
End if
15: End for
16: End for
17: End if
18: Else
19: Message1 = No error in pre-cycle.
18: End if
19: if $(i > n)$
*20:* for $k = 0\ to\ no$
*21:* if $\left(PostureFinal[k,0] = AsanaPosture\left[\frac{N}{2}+1\right]\right)$
22: $count++$
23: End if
24: End for
25: If$(count > 1)$
26: Message = Main posture not hold constantly
27: End if
28: End if
29: **if** $(i > \frac{n}{2}\ \ and\ i < n)$ **then**
30: Message1 = Error in Post-postures
31: Identify frame number of incorrect posture.

32: Store starting and ending frame numbers for incorrect postures sequence in $startFrame$ and $endFrame$ respectively.
33: $diff = endFrame - startFrame$
*34:* Message1 = *Āsana* posture not performed
*35:* Identify angle in frame which does not match with required posture angle
*36:* **for** $l = startFrame\ to\ endFrame$
*37:* **for** $n = 0\ to\ 10$
*38:* **If** $(PostureFinal[l, 7*n+m-1] = 0)$
39: IncorrectAngle[i] = $W2[l, 7*n+m-1]$
40: **End if**
41: **End for**
42: **End for**
43: **Else**
44: Message1 = No error in post-cycle.
45: **End if**
46: **End Procedure**

Step 29 to step 45 identifies error in pre-posture sequence in terms of posture and frame sequence and correct required angle values and measured angle values.

## 3.5. *Bharatnāṭyam Adavu* Recognition from Depth Data

*Bharatnāṭyam Adavu* Recognition System from Depth Data (BARSDD) recognizes the *Adavu* in *Bharatnāṭyam* using extended state model. *Bharatnāṭyam* is one of the Indian traditional dance forms. A specific sequence of poses in *Bharatnāṭyam* is known as *Adavu*. *Bharatnāṭyam* dances are based on stories of *Ramāyana* and *Mahābhārata,* the Indian Epics. Only, expert can understand this sentiment (*bhāva*) and posture (*mudrā*). Main motivation behind this work is that BARSDD system should recognize meaning of dance and display it for audience. *Adavu* is represented using extended state model. Five basic *Adavu Tatta*, *Natta*, *Sarikal*, *Visharu*, and *Kudittāmettā* are considered for recognition.

Focus of our work is identification of *Bharatnāṭyam Adavu*. *Bharatnāṭyam* is an Indian classical dance, having a rich history dating back to 2000 years. It was previously known as '*Dāssiattam*', due to classical performance by *Devdāsis*. It is fundamentally composed of *Bhāv* (sentiment), *Rāg* (musical theme), and *Tāl* (rhythmic pattern). Additionally, major and minor limb movements are also the part of the same. Such movements are called *Nritta*. The major limbs such as head, hands, legs and waist are essential parameters to recognize *Adavu*. Traditionally, *Bharatnāṭyam* transmits the story using hand gestures, facial expressions and body movements. However, the said approach is recognizable only by experts. Consequently, lay person has to exert more to understand the same. The research aims at recognition of story using body posture involving major limbs from camera captured contents. In order to have detail understanding of *Bharatnāṭyam Adavu* to the lay person, we have extended state model designed in our previous work.

### 3.5.1. *Bharatnāṭyam* Dataset Creation

Single pattern of five *Adavu* viz. *Tatta*, *Natta*, *Sarikal*, *Visharu*, and *Kudittāmettā* are considered in BARSDD system for recognition. The mentioned *Adavu* generate unique dataset. Here, three *Adavu Tatta*, *Natta*, and *Sarikal*, are discussed in detail, two *Adavu Visharu*, and *Kudittāmettā* are not given in detail, even though we used in experimentation. Kudittāmettā has seventeen key postures and *Visharu* has twenty eight key postures, which are very lengthy to represent. The detail representation is depicted in Figure 3.32. *Adavu* are captured using Kinect and obtained depth data of dancer is used for recognition. Depth data along with corresponding contour and state diagram of *Adavu* postures are shown in Figure 3.32. Depth data for Tatta, Natta and Sarikal are shown in Figure 3.32 (A), Figure 3.32 (D), and Figure 3.32 (G) respectively. Extracted contours from corresponding depth data are shown in Figure 3.32 (B), Figure 3.32 (E), and Figure 3.32 (H). Their state models are shown in Figure 3.32 (C), Figure 3.32 (F) and Figure 3.32 (I). In our previous work, state model is successfully applied for recognition of *Yogāsana* from video. Here, same is extended for recognition of *Bharatnāṭyam Adavu*. Each *Adavu* is composed of number of states. State is the prominent distinctive posture, hand symbol and/or leg position. Each state shows hand symbols (*Hasta Mudrā*), leg positions and body posture. Hand symbols includes *Pataka, KataKamukha, Alapadma, Suchi* etc. Leg position are *Udghatita, Samachaiva, Anchit* etc. Body postures are full sitting (*Mandali*), half sitting (*ArdhaMandali*) and standing position (*Stanaka Mandali*). Details of each *Adavu* are described as follows,

***Tatta:*** *Tatta* starts with flat foot contacts with ground in the basic *Ardhamandali* position and hands are enforced to be in star position at the back as shown in figure 3.32 (a1), figure 3.32 (a3), figure 3.32 (a5). Next, the posture is achieved with right and left leg elevated in succession and then the respective toe heading towards the ground as shown in Figure 4 (a2), figure 3.32 (a4). *Tatta Adavu* is composed of steps shown in figure 3.32 (a1...a5) performed repeatedly.

***Sarikal:*** This *Adavu* is performed by sliding or slipping sideways with both the feet in an erect position. Hands are anticipated in *Patākā* position with one hand near the chest and the other extended outwards parallel to the ground in the shoulder line with palm (*Patākā Hasta)* kept parallel to ground as shown in Figure 3.32 (b1) and 3.32 (b5). Afterwards the movements are materialized towards right with left leg sliding and palm intended towards the sky as shown in Figure 3.32 (b2) and figure 3.32 (b6). In next posture legs are portrayed in *Āñcita* position with *Patāka* palm pointing towards sky as shown in figure 3.32 (b3) and figure 3.32 (b7). Last position is feet in original position with *Patāka* palm pointing towards ground. *Sarikal Adavu* is composed of steps shown in Figure 3.32-(b1...b8) performed repeatedly.

**Natta:** It begins from *Ardhamandali* position and uses extension of legs as shown in figure 3.32 (c2) and figure 3.32 (c4). Hands are out stretched parallel to shoulders in *Tripatāka* position as shown in figure 3.32 (c1). The movement of leg along with

*Tripatāka* hand is shown in figure 3.32 (c1), figure 3.32 (c2), figure 3.32 (c3). It is observed that with the stretching of the leg *Tripataka* hand is bent downwards, followed by fingers pointing towards ground. *Natta Adavu* is composed of steps shown in figure 3.32 (c1...c5) performed repeatedly. A dancer generally repeat steps in *Adavu* starting with slow speed (*Vilambit*), medium speed (*Madhya*), and achieves fast speed (*Dritta*), again coming back to medium and then slow speed. System BARSDD can be represented by,

$SS$ = {$II$,, $FF$} (1)

Here, I - set of main input and intermediate inputs to system.

O - Set of main output and Intermediate outputs of system.

F - Set of functions

$II$ = {$II$1,2,$II$3,$II$4} (2)

$OO$ = {$OO$1,2,$OO$3,$OO$4,$OO$5} (3)

$FF$ = {$FF$1, $FF$2, $FF$3, $FF$4} (4)

F1 - Conversion of depth image sequence to Binary image

F2 - Finding Contour using CED Algorithm

F3 - Representation of contour using FCC (Feature vector representation)

F4 - Feature vector matching using Euclidean distance. Output of previous step is given as input to next step.

I1 - set of depth images captured using Kinect for *Adavu*

O1, I2 - set of binary images

O2, I3 - set of contour pixel values

O3, I4- set of connected chain code for respective contours.

O4, I5 - Matching results of testing data

O5 - recognized Adavu name.

$II1 = \{AA1, AA2, AA3, AA4, AA5\}$

Here, $AA1$, $AA2$, $AA3$, $AA4$, $AA5$ are set of *Tatta, Natta, Sarikala, Visharu and Kudittametta Adavu* depth frames respectively. $AAii \subseteq II1$ and it can be further represented as,

$II1 = \{a11, a12, \dots, a15\}, \{a21, a22, \dots, a25\}, \{a31, a32, \dots, a38\}, \{a41, a42, \dots, a427\}, \{a51, a52, \dots, a38\}$

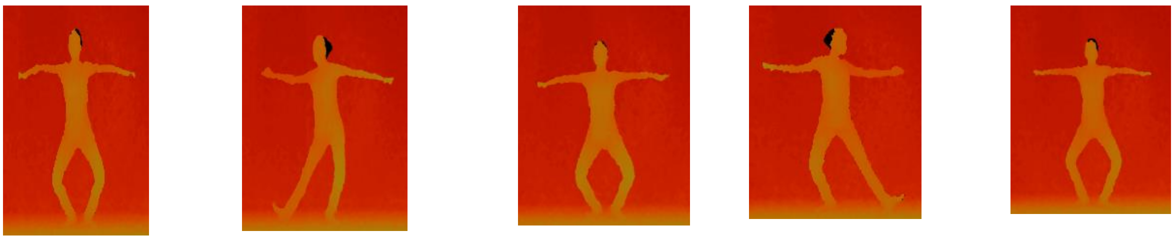
Here, *aij* is depth image. Where, *i* is *Adavu* number and *j* is frame number in particular *Adavu*. Identified distinct number of keyframes selected by observing dataset are considered for different *Adavu*. For *Tatta* and *Natta* five frames are considered, for *Sarikala* eight frames, *Kuddittametta* has seventeen frames and *Visharu* has twenty seven key frames considered. These depth frames are converted to binary images using F1. F2 extracts contour feature from each binary image using CED Algorithm. Here, main postures in Bharatnāṭyam are considered as states. Figure 3.32 (C), 3.32 (F), 3.32 (I) shows states considered in *Tatta, Sarikalā* and *Natta Adavu*. In this approach we need to design different state model for each *Adavu*.

*(a1)* *(a2)* *(a3)* *(a4)* *(a5)*

*(A): Depth image sequence for Tatta Adavu*

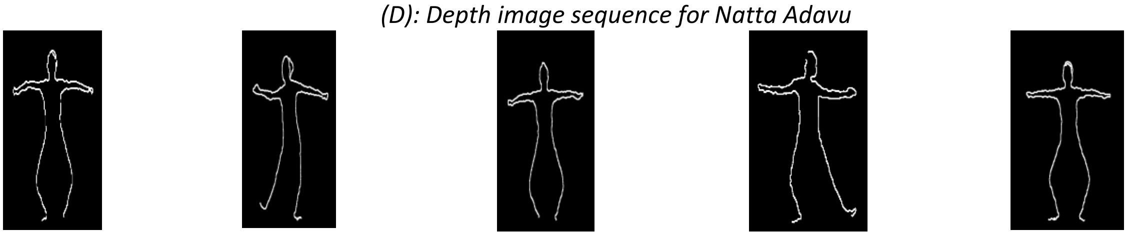
*(B): Corresponding contour for Tatta Adavu*


*(C): State Model for Tatta Adavu*



*(c1)* *(c2)* *(c3)* *(c4)* *(c5)*

*(D): Depth image sequence for Natta Adavu*



*(E): Corresponding contour for Natta Adavu*


*(F): State Model for Tatta Adavu*

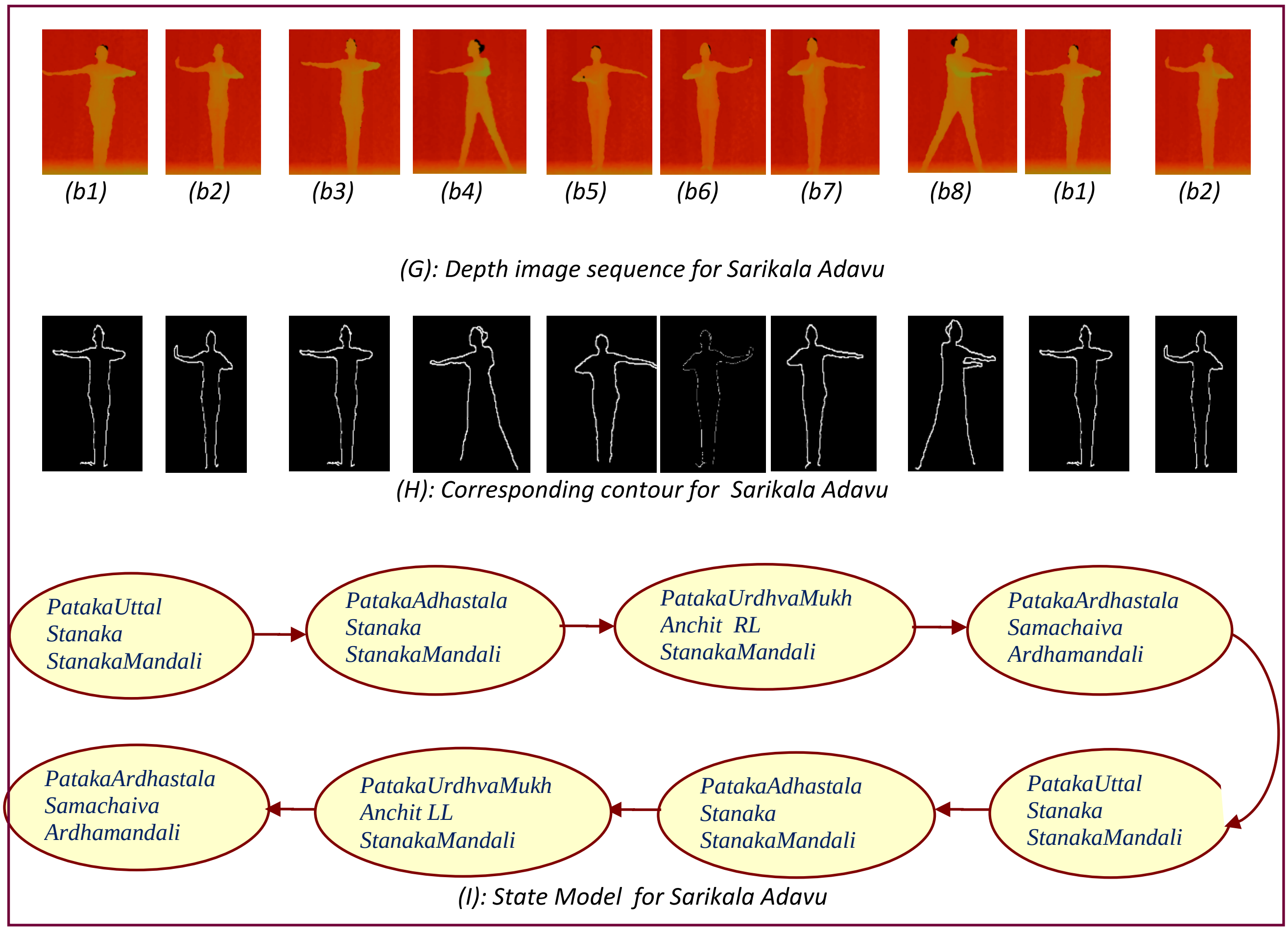



*Figure 3.32: Depth images, their corresponding contours and state model for Tatta, Natta, and Sarikala Adavu*

### 3.5.2. Methodology

*Adavu* i.e. sequence of specific posture from Indian traditional dance *Bharatnāṭyam* is recognized using BARSDD. Here, we considered contour feature of object for representation. The term contour can be defined as an outline or a boundary of an object. Further, the boundary points of contours have been extracted and encoded. Encoded features are used for matching. Match score is used to classify the *Adavu* in its corresponding class. Flow for BARSDD is shown in Figure 3.35. System accepts sequence of depth key frames of *Adavu* as input. These depth images are further converted to gray scale images in order to find out its contour. Canny Edge Detector (CED) is used for extraction of dancer's outline / contour from depth image. CED has several advantages over other edge detection algorithms. It is composed of steps:(i) image smoothing by Gaussian convolution, (ii) finding gradient, (iii) non-maximal suppression, (iv) double thresholding and (iv) tracking edge by hysteresis. CED detects discriminative edges for body parts. figure 1-a shows edge feature obtained for dancer's posture. Here, two contours for both hands and the main contour for whole body can be prominently seen along with small blobs at hair. But, our interest is in outer boundary of dancer's posture. To obtain this we have considered minimum and maximum X-values for each Y-value of every edge pixel. Obtained result is shown in Figure 1-b. Contour pixels are represented using Freeman Chain Code (FCC). FCC is a lossless compression algorithm for monochrome image contour representation. In FCC an arbitrary curve is represented by a

sequence of small vectors of unit length in 8 - direction as shown in Figure 3.34. The basic principle of chain codes is to encode separately each connected component in an image. For each such region, a point on the boundary is selected and its coordinates are the direction of this movement. This continues until the encoder returns to the starting position. To achieve high recognition rate, we apply FCC along with canny edge detection techniques. A chain code can be used to represent a binary image in a compact form, which can be complete representation of an object or a curve. Figure 3.35 shows flow of our system. Depth data of Bharatnāṭyam dancer is captured using Kinect sensor. Selected key frames are given as input for system. Frame contents are binarized for extraction of contour. Contour of human postures are then detected using Canny edge detection. Detected contours are represented using Directional Freeman Chain Code of 8 -Directions (DFCCE). Figure 3.34 shows direction labeling used. Both training and testing image contour features are represented using DFCCE. Feature vector is further compressed using run length encoding. represented by a sequence of small vectors of unit length in 8 -

*a. Result obtained using CED* *b. Modified result*

*Figure 3.33: Outer edge extraction of posture*

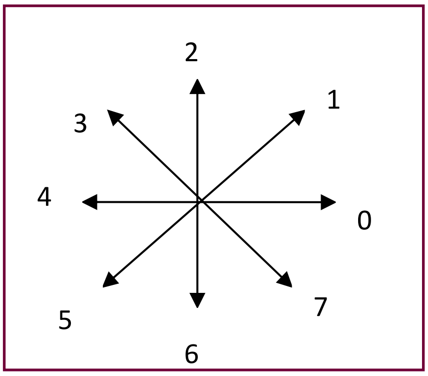



*Figure 3.34: Freeman chain code of 8 - directions*

direction as shown in Figure 3.34. The basic principle of chain codes is to encode separately each connected component in an image. For each such region, a point on the boundary is selected and its coordinates are the direction of this movement. This continues until the encoder returns to the starting position. The Freeman Chain Code technique gives commendable response time, but it compromises with accuracy. To achieve high recognition rate, we apply this technique along with CED techniques. A chain code can be used to represent a binary image in a compact form, which can be complete representation of an object or a curve. Flow of BARSDD system is shown in figure 3.35. Depth data of Bharatnāṭyam dancer is captured using Kinect sensor. Selected key frames are given as input for system. Frame contents are binarized for extraction of contour. Contour of human postures are then detected using canny edge detection. Detected contours are represented using Directional Freeman Chain Code of 8 - Directions (DFCCE). Figure 3.34 shows direction labeling used. Both training and testing image contour features are represented using DFCCE. Feature vector is further compressed using run length encoding. Recognition of *Adavu* is done by matching test data with training samples. Results are discussed in chapter 4.

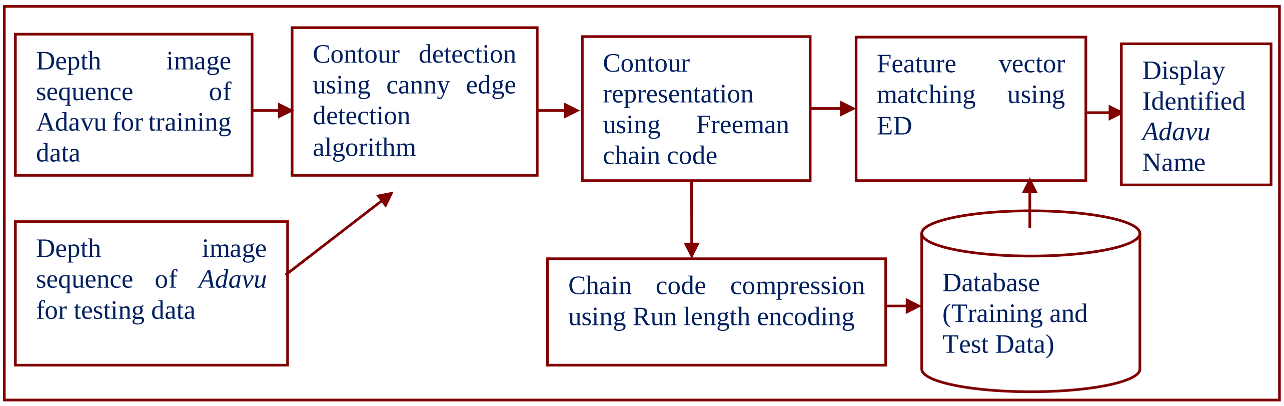


*Figure 4.35: Flow of BARSDD system*

*Yogāsana* action dataset is prepared and considered for corrective assessment and amendments are suggested for incorrect postures. As discussed in Literature Survey very few worked on problem of suggestion of amendment. Design of system along with algorithms is discussed in this chapter. Action execution speed invariant state transition model is designed and implemented to recognize *Yogāsana*. Three algorithms are designed; two for *Yogāsana* recognition and one for suggestion of amendments. First recognition algorithm considers real time data and uses joint positions as feature for recognition. Second algorithm considers offline dataset and uses angle feature for recognition. Third algorithm is designed to assess correctness in performed *Yogāsana* and provides required correction in practitioner's body postures. These algorithms use kinematic approach for representation of human postures, due to its benefits like camera view invariance, insensitivity to clothing etc. *Bharatnāṭyam Adavu* Recognition System from Depth Data (BARSDD) is designed and implemented using image model. From our experimentation we strongly recommend that kinematic features are more appropriate for suggestion of amendment in action.

# Chapter 4

# RESULTS AND ANALYSIS

# 4. Results and Analysis

*"If I could, I would always work in silence and obscurity, and let my efforts be known by their results."*

*-Emily Bronte*

e-*YogaGuru* system is designed using three layer hierarchical model and novel algorithms to suggest amendment in human action and specifically tested on created *Yogāsana* dataset. Suggestions for amendment are provided at an abstract level in terms of missing posture in the sequence of postures and detailed analysis is given in the form of an incorrect angle made by body parts. Results of the system with intermediate outcomes are discussed in this chapter.

## 4.1. Sensor Data and Intermediate Results

Dataset for experimentation is created using Kinect Xbox 360 initially and then with Kinect 1.0 after it was released. Kinect sensor is cost effective as compared to available cameras like ToF and its performance is almost same as ToF [31] - [34].

For experimentation we have taken input from skeleton tracking of Kinect sensor. Experimentation is done in laboratory indoor environment. *Yogāsana* dataset is prepared on physically fit 20 to 25 years old individuals. Sample values of joint positions for *Ūrdhva Hāstāsana* are shown in table 4.1. Angle features are used for representation of action and sample body angle features computed from joint points using algorithm 3 are shown in table 4.2.

Performance of Kinect Xbox 1.0 in terms of skeleton joint position accuracy, stability of data and with more joint positions is increased. This helped to increase performance of algorithms that designed at higher level by taking this raw data as input. Angle features for *Ūrdhva Hāstāsana* computed from skeleton data captured using Kinect 360 and Kinect 1.0

*Table 4.1: Data captured with Kinect 1.0 (X, Y) co-ordinate*

| Frame No. | $X_1$ | $Y_1$ | $X_2$ | $Y_2$ | $X_3$ | $Y_3$ | $X_4$ | $Y_4$ | $X_5$ | $Y_5$ | | $X_{24}$ | $Y_{24}$ | $X_{25}$ | $Y_{25}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **7400** | 244 | 125 | 245 | 156 | 223 | 163 | 268 | 160 | 215 | 194 | ... | 303 | 244 | 271 | 152 |
| **7425** | 243 | 125 | 245 | 156 | 222 | 164 | 267 | 160 | 215 | 194 | ... | 303 | 244 | 270 | 152 |
| **7450** | 243 | 125 | 245 | 156 | 222 | 164 | 267 | 160 | 215 | 194 | ... | 303 | 244 | 270 | 152 |
| **7475** | 243 | 125 | 245 | 156 | 222 | 164 | 267 | 160 | 214 | 194 | ... | 303 | 245 | 270 | 152 |
| **7500** | 243 | 125 | 245 | 156 | 222 | 164 | 267 | 160 | 214 | 194 | ... | 303 | 244 | 270 | 152 |
| **7525** | 243 | 125 | 245 | 156 | 222 | 164 | 267 | 160 | 214 | 194 | ... | 303 | 244 | 270 | 152 |
| **7550** | 243 | 125 | 245 | 156 | 222 | 164 | 267 | 160 | 214 | 194 | ... | 303 | 244 | 270 | 152 |
| **7575** | 243 | 125 | 245 | 156 | 222 | 164 | 267 | 160 | 214 | 194 | ... | 303 | 244 | 270 | 152 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ... | ⋮ | ⋮ | ⋮ | ⋮ |
| **7850** | 244 | 125 | 245 | 156 | 222 | 163 | 267 | 160 | 214 | 194 | ... | 269 | 226 | 244 | 145 |
| **7875** | 244 | 125 | 245 | 156 | 222 | 163 | 267 | 160 | 214 | 195 | ... | 270 | 226 | 244 | 145 |
| **7900** | 244 | 124 | 245 | 156 | 222 | 163 | 267 | 160 | 214 | 195 | ... | 270 | 226 | 244 | 145 |
| **7925** | 244 | 124 | 245 | 156 | 221 | 163 | 267 | 160 | 214 | 195 | ... | 270 | 226 | 244 | 145 |
| **7950** | 244 | 124 | 245 | 156 | 221 | 163 | 267 | 160 | 214 | 195 | ... | 269 | 226 | 244 | 145 |
| **7975** | 244 | 124 | 245 | 156 | 221 | 163 | 267 | 160 | 214 | 194 | ... | 269 | 225 | 244 | 145 |
| **8000** | 244 | 124 | 245 | 156 | 221 | 163 | 267 | 160 | 214 | 195 | | 269 | 225 | 244 | 145 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ... | ⋮ | ⋮ | ⋮ | ⋮ |

*Table 4.2: Body angle values computed using Kinect 1.0 data*

| Frame No. | $A_1$ | A2 | $A_3$ | $A_4$ | $A_5$ | $A_6$ | $A_7$ | $A_8$ | $A_9$ | $A_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| **7400** | 139 | 114 | 117 | 179 | 175 | 179 | 95 | 93 | 174 | 175 |
| **7425** | 139 | 114 | 117 | 178 | 175 | 179 | 95 | 93 | 174 | 175 |
| **7450** | 139 | 114 | 117 | 178 | 175 | 179 | 95 | 93 | 174 | 175 |
| **7475** | 139 | 114 | 117 | 178 | 175 | 179 | 95 | 93 | 174 | 175 |
| **7500** | 139 | 115 | 117 | 178 | 175 | 179 | 95 | 93 | 174 | 175 |
| **7525** | 139 | 114 | 117 | 178 | 175 | 179 | 95 | 93 | 174 | 175 |
| **7550** | 139 | 114 | 117 | 178 | 175 | 179 | 95 | 93 | 174 | 175 |
| **7575** | 140 | 114 | 117 | 178 | 174 | 179 | 95 | 93 | 174 | 175 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ... | ⋮ | ⋮ | ⋮ | ⋮ |
| **7850** | 142 | 142 | 142 | 197 | 172 | 179 | 88 | 153 | 184 | 48 |
| **7875** | 142 | 142 | 142 | 197 | 172 | 179 | 88 | 153 | 184 | 43 |
| **7900** | 142 | 142 | 142 | 203 | 172 | 179 | 88 | 153 | 184 | 43 |
| **7925** | 142 | 142 | 142 | 197 | 172 | 179 | 88 | 159 | 184 | 41 |
| **7950** | 142 | 142 | 142 | 197 | 172 | 179 | 88 | 155 | 184 | 42 |
| **7975** | 142 | 142 | 142 | 197 | 172 | 179 | 88 | 155 | 183 | 43 |
| **8000** | 142 | 142 | 142 | 198 | 173 | 179 | 88 | 154 | 184 | 43 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ... | ⋮ | ⋮ | ⋮ | ⋮ |

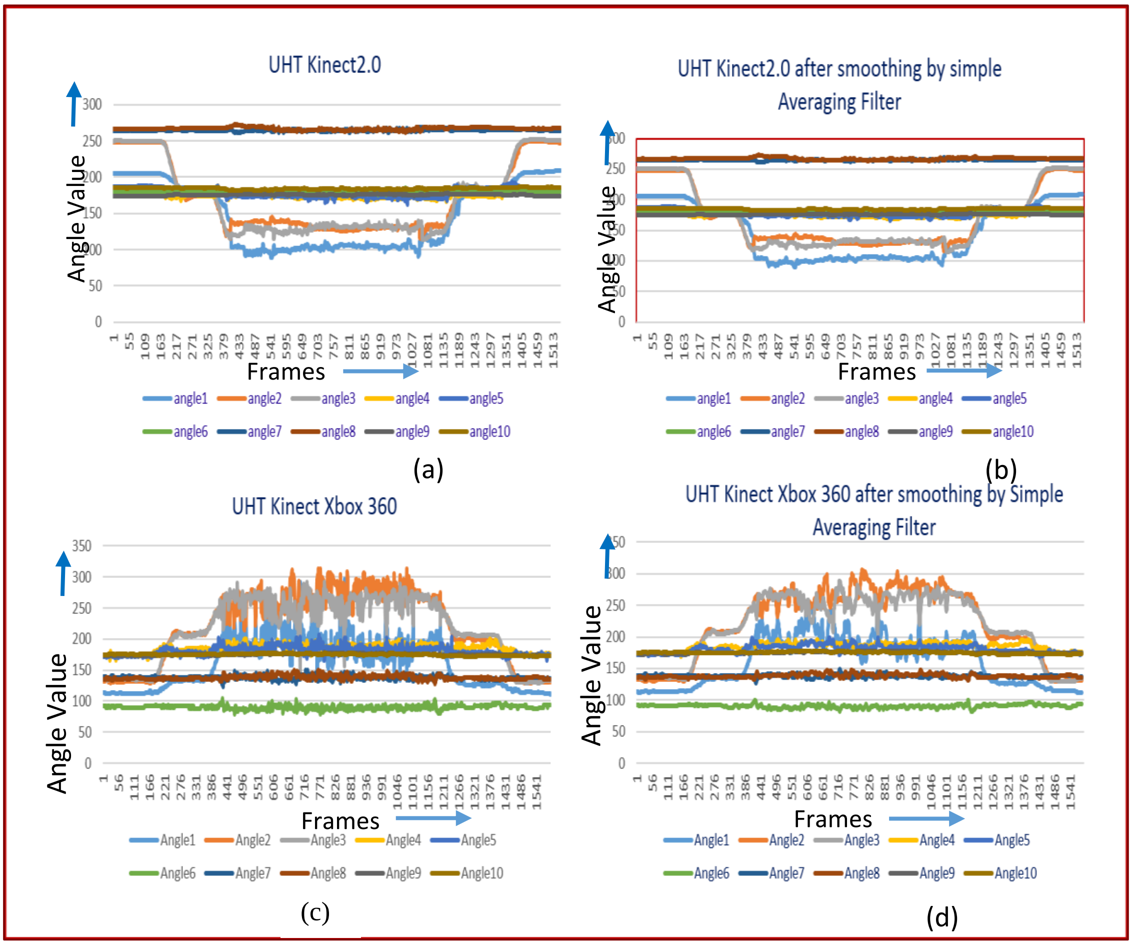



*Figure 4.1: Comparison of human body angle values obtained using Kinect 360 and Kinect 1.0*

are shown in figure 4.1 (a) and figure 4.1 (c) respectively. We can observe more spikes in features computed from data captured using Kinect Xbox 360 as shown in figure 4.1 (c) as compared to Kinect 1.0 shown in figure 4.1 (a).

Obtained data is further smoothened using Simple Moving Averaging Filter to reduce spikes in data and smoothened results are shown in figure 4.1(b) and figure 4.1(d).

Raw sensor data is provided as input for designed algorithms, its results are discussed and analyzed in subsequent sections.

## 4.2. Real Time *Yogāsana* Recognition System using Joint Feature

System is tested on three *Yogāsana* - *Samasthiti Tādāsana*, *Ūrdhva Hāstāsana*, *Vīrabhadrāsana-II*. It is assumed that practitioner knows the *Yogāsana* and tries to perform it in front of camera. X and Y co-ordinates of twenty joint positions of human body obtained from Kinect are considered as feature for recognition. All three

*Yogāsana* are tested on four female subjects and each *Yogāsana* is repeated 10 times i.e. 3 × 4 × 10 = 120 video sequences. Table 4.2 shows results obtained for three *Yogāsana* repeated 40 times. Algorithm gives 100% results for tested data and being rule based system, does not require training.

*Table 4.3: (a) Recognition results for real time Yogāsana using joint value (b) Details of posture labels*

***(a)***

Predicted Class

| True Class | ST | UHT | VB |
|---|---|---|---|
| ST | 40 | | |
| UHT | | 40 | |
| VB | | | 40 |

***(b)***

| Gesture | Description |
|---|---|
| ST | *Samasthiti Tādāsana* |
| UHT | *Ūrdhva Hāstāsana* |
| VB | *Vīrbhadrāsana* |

It is observed that joint features used by discussed approach are not convenient for suggestion of amendment, so angle features are considered for representation of human postures. Also, joint features are sensitive to human anthropometry as compared to angle features. Details are discussed in section 3.1.3.

## 4.3. Two Layer Hierarchical Model for Human Motion Recognition

*Yogāsana* is comparatively long video sequence and it has multiple postures in one sequence, so two layered hierarchical model is used for recognition. Posture in individual frame is recognized at lower layer and *Yogāsana* is recognized at higher layer. Results of posture recognition at lower layer are obtained using ED between feature values of postures in each frame and postures in knowledge base prepared in consultation with trainer using expert's data. *Yogāsana* is recognized at higher layer using state transition model designed for recognition of *Yogāsana* discussed in chapter 3 in detail. Results of recognition of *Ūrdhva Hāstāsana* and *Vīrabhadrāsana-II* are discussed in this section. Only two *Yogāsana* are considered for discussion as main focus of research is suggestion of amendment in incorrect postures and recognition results are intermediate results.

*Table 4.4: (a) Recognition results for Yogāsana postures using angle feature, (b) Details of posture labels (Recognition classes)*

**(a)**

Predicted Class

| True Class | a | c | d | g | h | i |
|---|---|---|---|---|---|---|
| **a** | 13290 | | | | | |
| **c** | | 4677 | | | | |
| **d** | | | 9810 | | | |
| **g** | | | | 3612 | | |
| **h** | | | | | 3865 | 73 |
| **i** | | | | | | 6376 |

**(b)**

| Gesture No | Description |
|---|---|
| a | *Samasthiti* |
| c | *Ūrdhva Hāstāsana Hands Apart* |
| d | *Hands Up* |
| g | *Vīrbhadrāsana Legs Apart* |
| h | *Vīrbhadrāsana Hands and Legs Apart* |
| i | *Knee Bend ( Vīrbhadrāsana Main posture)* |

Table 4.4 (a) shows confusion matrix for postures of *Ūrdhva Hāstāsana* and *Vīrabhadrāsana-II* at lower layer and posture label description is shown in table 4.4 (b). It is observed that all the postures are identified correctly except *Vīrabhadrāsana-II* main posture, as there is very less difference between posture *'h'* and posture *'i'* in terms of angle value. Out of 6449 frames with *Vīrabhadrāsana-II* main posture system is confused for 73 frames and they are identified as posture hands and legs apart i.e. 'h'.

Posture recognition result obtained at lower layer is provided as input for higher layer for recognition of *Yogāsana*. Required posture sequence for *Ūrdhva Hāstāsana* is a-c-d-c-a and for *Vīrabhadrāsana-II* is a-g-h-i-h-g-a. Column one of table 4.5 (a) and table 4.5 (b) represents posture labels for *Ūrdhva Hāstāsana* and *Vīrabhadrāsana-II* respectively. Column labeled A1 to A20 represents number of frames for all postures in particula*r Yogāsana* iteration. Each row talks about the identified number of frames with labeled correct postures. Here, correct sequence identified in each column ensures the type of *Yogāsana* performed. All performed *Yogāsana* are recognized correctly.

*Table 4.5: Recognition results for (a) Ūrdhva Hāstāsana (b) Vīrabhadrāsana-II*

***(a)***

| Asana / posture | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | A10 | A11 | A12 | A13 | A14 | A15 | A16 | A17 | A18 | A19 | A20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **a** | 116 | 116 | 130 | 162 | 174 | 144 | 62 | 57 | 248 | 154 | 348 | 344 | 278 | 390 | 200 | 114 | 364 | 190 | 272 | 336 |
| **g** | 61 | 102 | 50 | 99 | 67 | 76 | 65 | 62 | 60 | 76 | 134 | 106 | 90 | 127 | 125 | 131 | 133 | 113 | 81 | 160 |
| **h** | 0 | 96 | 94 | 58 | 63 | 111 | 74 | 79 | 50 | 127 | 146 | 93 | 145 | 190 | 142 | 111 | 138 | 106 | 77 | 98 |
| **i** | 374 | 229 | 216 | 170 | 161 | 118 | 158 | 172 | 142 | 165 | 276 | 330 | 328 | 371 | 505 | 501 | 401 | 433 | 538 | 783 |
| **h** | 84 | 47 | 38 | 46 | 50 | 122 | 48 | 26 | 71 | 105 | 220 | 162 | 131 | 112 | 138 | 110 | 84 | 112 | 96 | 57 |
| **g** | 50 | 60 | 54 | 47 | 75 | 51 | 60 | 68 | 72 | 83 | 131 | 117 | 101 | 123 | 94 | 101 | 105 | 60 | 101 | 135 |
| **a** | 112 | 78 | 168 | 59 | 83 | 25 | 97 | 57 | 143 | 52 | 87 | 126 | 126 | 83 | 134 | 132 | 201 | 157 | 125 | 158 |

***(b)***

| Asana / posture | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | A10 | A11 | A12 | A13 | A14 | A15 | A16 | A17 | A18 | A19 | A20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **a** | 126 | 78 | 110 | 134 | 112 | 70 | 74 | 132 | 114 | 376 | 260 | 290 | 358 | 208 | 360 | 394 | 296 | 226 | 178 | 260 |
| **c** | 82 | 71 | 72 | 70 | 26 | 71 | 34 | 77 | 47 | 67 | 161 | 121 | 137 | 75 | 127 | 116 | 148 | 242 | 448 | 161 |
| **d** | 301 | 296 | 299 | 240 | 324 | 269 | 290 | 289 | 257 | 260 | 971 | 498 | 536 | 519 | 765 | 677 | 763 | 712 | 573 | 971 |
| **c** | 61 | 82 | 61 | 87 | 62 | 69 | 97 | 83 | 75 | 64 | 226 | 92 | 152 | 159 | 170 | 147 | 160 | 123 | 128 | 226 |
| **a** | 128 | 81 | 52 | 227 | 153 | 88 | 84 | 96 | 143 | 126 | 186 | 147 | 145 | 130 | 123 | 101 | 148 | 296 | 92 | 186 |

Following are the observations and analysis from experimentation of two methodologies discussed in section 4.2 and 4.3

- From table 4.5 (a) and table 4.5 (b) it is observed that number of frames with particular posture varies for each *Yogāsana* iteration. It is due to same practitioner may take different amount of time for same *Yogāsana* in different iteration. Other recognition approaches like template matching may fail for recognition, as there is no exact match of sequences with respective time. DTW helps for time invariant recognition, but it may fail for suggestion of amendment. Designed and implemented modified state model approach serves both purposes i.e. time invariance and suggestion of amendment.

- Designed state model with knowledge-base provides reduced training data.

- Use of angle feature for representation of body reduced feature vector from 40 dimensional joint positions to 10 dimensional angle values. Angle features also ensures anthropometric invariance as body angles are less sensitive to anthropometry as compared to motion trajectories. Also, angle features are more accurate and discriminative.

- However, consideration of only angle features is not suitable for actions like body up, body down etc.
- Angle feature vector matching shows success in action recognition, but fails in suggestion of amendment in human action, as it does not work on individual angle position. Euclidian distance between summations of body angle cannot distinguish between *Vīrabhadrāsana -II* done with left and right leg.

To overcome above mentioned issues individual angles are verified from knowledge-base for recognition and suggestion of amendment.

## 4.4. Three Layer Hierarchical System for Suggesting Amendment

Practitioner may perform erroneous posture while practicing *Yogāsana* and it is very important to correct it in order to avoid adverse effect. Algorithms designed and tested for recognition of *Yogāsana* provide foundation for suggestion of amendment. Designed and implemented system e-YogaGuru identifies correctness of *Yogāsana*. It suggests amendment in erroneous postures and also identifies missing posture in posture sequence. Results of designed algorithms are experimented and discussed for seven *Yogāsana*. Sample common errors are discussed for each *Yogāsana.* Practitioner either do not perform particular posture or he or she may perform it erroneously. System identifies both the cases and displays details at two levels i.e. abstract level and detail amendment. There can be single or multiple errors in *Yogāsana*. If there are multiple errors first error in pre-posture sequence is analyzed and displayed, then main posture error is analyzed and displayed and finally post-posture sequence is analyzed and error is displayed. Results for classification as correct and erroneously performed *Yogāsana*, abstract level amendment and detailed amendment for sample cases are discussed in this section.

Dataset is created on 25 practitioners with different height and weight, details are as shown in figure 4.2. Most probable errors observed by *Yogāsana* trainer are discussed here, more cases of errors may exist.

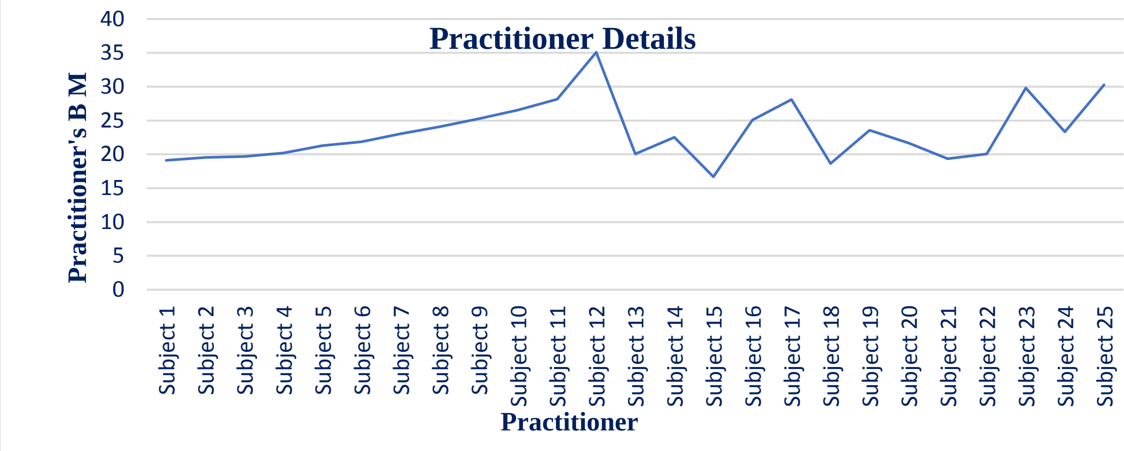



*Figure 4.2. Practitioner's details - subject Vs BMI*

### *4.4.1. Samasthiti Tādāsana*

Sample seven cases of *Samasthiti Tādāsana* are shown in table 4.6. Out of that first is correct *Samasthiti Tādāsana* and remaining are erroneous *Samasthiti Tādāsana*. Few erroneous cases are discussed in detail for *Samasthiti Tādāsana.*

*Table 4.6: Possible sample sequences for Samasthiti Tādāsana*

| Sr. No. | Sequence | Details |
|---|---|---|
| 1 | $ab_1ba$ | Correctly done Asana. |
| Single Error in *Ūrdhva Hāstāsana* | | |
| 2 | $ab_1a$ | Standing on toes pose not hold for 10 sec./ 3 breathings. |
| 3 | $a\ b_1a\ b_1a$ | Toes taken down In between along with count. |
| 4 | ab | Erroneous / Missing pose - Samasthiti toes down in post-postures. |
| 5 | ba | Erroneous / Missing pose - Samasthiti toes down in pre-postures. |
| 6 | a | Erroneous / Missing pose - toes up. |
| 7 | - | Samasthiti Tādāsana not identified. |

In result tables, first column represents posture label, second column represents number of times corresponding posture appeared in video and third column represents posture hold time in seconds. All result tables discussed here represents output generated by e-*YogaGuru* and same conventions should be monitored throughout while reading result tables.

Table 4.7 shows recognition results for correctly performed *Samasthiti Tādāsana.* All three postures are identified in correct expected sequence and main posture i.e. *Samasthiti* body up is hold for 21.96 seconds.

*Table 4.7: Correctly done Samasthiti Tādāsana*

| Actual recognized sequence is: | | |
|---|---|---|
| a | 129 | 4.3 sec. |
| b | 6588 | 21.96 sec. |
| a | 120 | 4 sec. |
| Samasthiti Tādāsana performed correctly. You are doing well.... Keep it up!! | | |
| Current hold time is: 21.96667sec. | | |

Most probable error while performing *Samasthiti Tādāsana* is practitioner may take legs down while performing '*Samasthiti* body up main posture'. That means practitioner can't balance body weight on toes. *Yogāsana* detail shown in table 4.8 is example of toes taken down in between. Practitioner may take legs down in number of times while performing *Yogāsana* and again take legs up to adjust with pose, encircled with red. So, count of number of times legs taken down is also shown in results. Here, legs are taken down only once.

*Table 4.8: Toes taken down in between along with count*

| Actual recognized sequence is: | | |
|---|---|---|
| a | 40 | 1.333 Sec. |
| b | 16 | 0.533 Sec. |
| a | 124 | 4.133 Sec. |
| b | 1013 | 33.766 Sec. |
| a | 234 | 7.8 Sec. |
| | | |
| Standing on toes pose is not hold constantly. Toes are taken down | | |
| Number of times Toes are taken down : 1 | | |

It may happen that practitioner did not start with correct Samasthiti posture and afterwards adjust body for correct posture. Case is shown in table 4.9. First 368 frames have identified incorrect postures and after that postures are identified correctly. At abstract level result is displayed as, "Wrongly performed Samasthiti Tādāsana." After that correct sequence of postures that needs to be performed is displayed. In detailed amendment frame numbers with wrong postures are displayed.

*Table 4.9: Samasthiti legs down missing in Pre- cycle*

| Actual recognized sequence is: | | |
|---|---|---|
| b | 354 | 11.8 Sec. |
| a | 117 | 3.9 Sec. |
| | | |
| Wrongly performed Samasthiti Tādāsana | | |
| Following sequence of postures should be performed: | | |
| a Samasthiti | | |
| b Samasthiti body up | | |
| a Samasthiti | | |
| Error after frame : | 0 | 29215 |
| Error before frame | 368 | 29583 |

Table 4.10 shows results obtained for *Samasthiti Tādāsana* with error - legs down missing in post-cycle.

*Table 4.10: Samasthiti or legs down missing in post-cycle*

| Actual recognized sequene is: | | |
|---|---|---|
| a | 84 | 2.8 Sec. |
| b | 1021 | 34.033 Sec. |
| | | |
| Wrongly performed Samasthiti Tādāsana | | |
| Body is not taken down | | |
| | | |
| Initially performed correct Asana postures are | | |
| a Samasthiti | | |
| b Samasthiti Body Up | | |
| after that following sequence of postures should be performed : | | |
| a Samasthiti | | |

Table 4.11 (a) shows case where *Samasthiti Tādāsana* is performed with wrong hand positions. Practitioner is asked to keep his hands in wrong position and perform *Samasthiti Tādāsana*. All angles except $\angle 2$ *and* $\angle 3$ identified correctly. At abstract level it is identified that all postures in *Samasthiti Tādāsana* are erroneous and error is identified from frame number 26597 to 277121. In detailed amendment angle details are displayed

*Table 4.11 (a): Samasthiti performed with wrong hand positions. No Samasthiti Tādāsana postures recognized (Abstract Level Suggestion for Amendment)*

| | | |
|---|---|---|
| Actual recognized sequence is: | | |
| 0 | 0 | 0 Sec. |
| | | |
| Wrongly performed Samasthiti Tādāsana | | |
| | | |
| Following sequence of postures should be performed : | | |
| a Samasthiti | | |
| b Samasthiti Body Up | | |
| a Samasthiti | | |
| | | |
| Error after frame : | 0 | 26597 |
| Error before frame : | 1023 | 27712 |

with frame number, angle number, expected range of angle and recorded angle at that position shown in table 4.11 (b).

*Table 4.11 (b): Angle details for erroneous Samasthiti performed with wrong hand positions. No Samasthiti Tādāsana postures recognized* (Detailed suggestion for amendment)

| Frame Number | **Angle NO** | Required Min | Required Max | Actual Recorded | **Angle NO** | Required Min | Required Max | Actual Recorded |
|---|---|---|---|---|---|---|---|---|
| 26597 | 2 | 137 | 147 | 192 | 3 | 134 | 144 | 190 |
| 26601 | 2 | 137 | 147 | 192 | 3 | 134 | 144 | 192 |
| 26602 | 2 | 137 | 147 | 192 | 3 | 134 | 144 | 190 |
| 26603 | 2 | 137 | 147 | 192 | 3 | 134 | 144 | 190 |
| 26604 | 2 | 137 | 147 | 192 | 3 | 134 | 144 | 192 |
| 26605 | 2 | 137 | 147 | 194 | 3 | 134 | 144 | 192 |
| 26606 | 2 | 137 | 147 | 194 | 3 | 134 | 144 | 193 |
| 26608 | 2 | 137 | 147 | 193 | 3 | 134 | 144 | 192 |
| 26609 | 2 | 137 | 147 | 194 | 3 | 134 | 144 | 192 |
| 26610 | 2 | 137 | 147 | 194 | 3 | 134 | 144 | 189 |
| 26611 | 2 | 137 | 147 | 194 | 3 | 134 | 144 | 189 |
| 26612 | 2 | 137 | 147 | 194 | 3 | 134 | 144 | 192 |
| . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . |

### *4.4.2. Ūrdhva Hāstāsana*

Sample fifteen cases of Ūrdhva Hāstāsana are discussed in table 4.12, out of that first is correct *Ūrdhva Hāstāsana* and remaining are erroneous *Ūrdhva Hāstāsana*. Table 4.13 shows recognition results for correctly performed *Ūrdhva Hāstāsana*. Here, all seven postures are identified in correct expected sequence and main posture i.e. *Ūrdhva Hāstāsana* body up is hold for 54.266 seconds.

*Table 4.12: Possible sample sequences for Ūrdhva Hāstāsana.*

| Sr. No. | Sequence | Details |
|---|---|---|
| 1 | acdedca | Correctly done *Āsana* |
| Single Error in *Ūrdhva Hāstāsana* | | |
| 2 | acde$_1$edca | Standing on toes pose not hold for 10 sec./ 3 breathings. |
| 3 | acde**dede**dca | Toes taken down In between along with count. |
| 4 | ad e$_1$edca | Erroneous / Missing pose – Hands Apart in Pre-posture |
| 5 | acd e$_1$eda | Erroneous / Missing pose – Hands Apart in Post-posture |
| 6 | cdedca | Erroneous / Missing pose – Samasthiti in Pre-posture |
| 7 | acdedc | Erroneous / Missing pose – Samasthiti in Post-posture |
| 8 | acdca | Erroneous / Missing pose – Standing on Toes |
| 9 | acedca | Erroneous / Missing pose – Hands Up in Pre-posture |
| 10 | acdeca | Erroneous / Missing pose – Hands Up in Post-posture |
| 11 | acdedcaca | Extra movements are done after *Yogāsana* |
| Multiple Errors in *Ūrdhva Hāstāsana* | | |
| 12 | aca | Hands Up Not Done |
| 13 | acdea | Erroneous / Missing pose – Hands up and Hands apart in Post-posture |
| 14 | aedca | Erroneous / Missing pose – Hands Apart and Hands up in Pre-posture |
| 15 | adededc | **Pre-posture:** Hands Apart Missing<br>**Main Posture**: Foots taken down<br>**Post-posture:** *Samasthiti* Missing |

*Table 4.13: Correctly done Ūrdhva Hāstāsana*

| Actual recognized sequence is: | | |
|---|---|---|
| a | 451 | 15.033 Sec. |
| c | 129 | 4.3 Sec. |
| d | 255 | 8.5 Sec. |
| e | 1628 | 54.266 Sec. |
| d | 519 | 17.8 Sec. |
| c | 337 | 11.233 Sec. |
| a | 103 | 3.433 Sec. |
| *Ūrdhva Hāstāsana* performed correctly.<br>You are doing well… Keep it up!! | | |

While performing any *Yogāsana*, main posture is expected to hold for at least ten seconds or three breathings. Ten seconds are considered for novice practitioner and regular practitioner should hold it 30 seconds to 1 minute. Table 4.14 shows case where *Ūrdhva*

*Table 4.14: Standing on toes pose not hold for sufficient time in Ūrdhva Hāstāsana*

| Actual recognized sequence is : | | |
|---|---|---|
| a | 275 | 9.166 Sec. |
| c | 32 | 1.066 Sec. |
| d | 12 | 0.4 Sec. |
| e | 151 | 5.033 Sec |
| d | 188 | 6.266 Sec. |
| c | 15 | 0.5 Sec. |
| a | 117 | 3.9 Sec. |
| | | |
| Increase Hold Time of standing on toes state to 10 Sec / 3 breathings. Current hold time is: 5.033333 sec. | | |

*Hāstāsana* body up posture not hold for required amount of time. Hold time is 5.033 seconds.

Another most probable error while performing *Ūrdhva Hāstāsana* is practitioner may take legs down while performing *Ūrdhva Hāstāsana* body up main posture. That means practitioner can't balance body weight on toes. *Yogāsana* detail shown in table 4.16 is an example of toes taken down in between. Practitioner may take legs down number of times while performing *Yogāsana* and again take legs up to adjust with pose. So, count of number of times legs taken down is also shown in results. Here, legs are taken down twice.

*Table 4.15: Standing on toes pose is not hold constantly in Ūrdhva Hāstāsana*

| Actual recognized sequence is : | | |
|---|---|---|
| a | 191 | 6.3666 Sec. |
| c | 99 | 3.3 Sec. |
| d | 301 | 10.03 Sec. |
| e | 180 | 6 Sec. |
| d | 42 | 1.4 Sec. |
| e | 744 | 24.8 Sec. |
| d | 24 | 0.8 Sec. |
| e | 597 | 19.9 Sec. |
| d | 307 | 10.23 Sec. |
| c | 137 | 4.56 Sec. |
| a | 84 | 2.8 Sec. |
| Wrongly performed Ūrdhva Hāstāsana | | |
| | | |
| Initially performed correct Asana postures are | | |
| a Samasthiti | | |
| c Hands Apart | | |
| d Hands Up | | |
| e Hands Up body Up | | |
| d Hands Up | | |
| After that following sequence of postures should be performed : | | |
| c Hands Apart | | |
| a Samasthiti | | |
| Standing on toes pose is not hold constantly. Foots are taken down between 2 Times | | |

Table 4.16 discuss case of erroneous posture hands apart in pre-posture. Here, practitioner has performed erroneous hands apart posture in pre-posture sequence. System has identified it and displayed correct expected sequence in abstract level suggestion of recorded angle value and identified angle value for $\angle 2\ and\ \angle 3$.

*Table 4.16: Amendment in Ūrdhva Hāstāsana with erroneous posture hands apart in pre-posture*

| Actual recognized sequence is: | | | |
|---|---|---|---|
| a | 284 | 9.3466 Sec. | |
| d | 134 | 4.466 Sec. | |
| e | 1575 | 52.5 Sec. | |
| d | 112 | 37.33 Sec. | |
| c | 127 | 42.33 Sec. | |
| a | 39 | 13 Sec. | |
| | | | |
| Abstract Level Analysis: | | | |
| Wrongly performed *Ūrdhva Hāstāsana* | | | |
| Initially performed correct *Asana* postures are | | | |
| a Samasthiti | | | |
| After that following sequence of postures should be performed : | | | |
| c Hands Apart | | | |
| Finally performed correct Asana postures are | | | |
| d Hands Up | | | |
| e Hands Up body Up | | | |
| d Hands Up | | | |
| c Hands Apart | | | |
| a Samasthiti | | | |
| Error in pre-posture | | | |
| | | | |
| Detailed Analysis: | | | |
| Error after frame : | 285 | 14388 | |
| Error before frame : | 441 | 14544 | |
| | | | |
| Erroneous angles are: | | | |
| **Angle** | **Required Min** | **Required Max** | **Actual Recorded** |
| 2 | 215 | 225 | 153 |
| 3 | 215 | 225 | 154 |

*Table 4.17: Angle details for erroneous Ūrdhva Hāstāsana with erroneous posture hands apart in pre-posture*

| Frame No | Angle No | Required Min | Required Max | Actual Recorded | Frame No | Angle No | Required Min | Required Max | Actual Recorded |
|---|---|---|---|---|---|---|---|---|---|
| 14388 | 2 | 218 | 228 | 150 | 14388 | 3 | 210 | 220 | 150 |
| 14389 | 2 | 218 | 228 | 154 | 14389 | 3 | 210 | 220 | 154 |
| 14390 | 2 | 218 | 228 | 152 | 14390 | 3 | 210 | 220 | 154 |
| 14391 | 2 | 218 | 228 | 153 | 14391 | 3 | 210 | 220 | 153 |
| 14392 | 2 | 218 | 228 | 154 | 14392 | 3 | 210 | 220 | 154 |
| 14393 | 2 | 218 | 228 | 153 | 14393 | 3 | 210 | 220 | 153 |
| 14394 | 2 | 218 | 228 | 153 | 14394 | 3 | 210 | 220 | 154 |
| 14395 | 2 | 218 | 228 | 152 | 14395 | 3 | 210 | 220 | 152 |
| 14396 | 2 | 218 | 228 | 153 | 14396 | 3 | 210 | 220 | 154 |
| . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . |

amendment and in detailed analysis frame sequence with erroneous postures is identified and displayed. Further it is identified that ∠2 and ∠3 has error and table 4.17 shows details of actual recorded angle for practitioners posture and correct expected range of same angle frame wise along with frame number.

### *4.4.3. Vīrabhadrāsana-II*

Sample twelve cases of *Vīrabhadrāsana-II* are discussed in table 4.18, out of that first is correct *Vīrabhadrāsana-II* and remaining are erroneous *Vīrabhadrāsana-II.*

*Table 4.18: Possible sample sequences for Vīrabhadrāsana-II*

| Sr. No. | Sequence | Details |
|---|---|---|
| 1 | *a g h il h g a* | Correctly Done Asana with left knee bend. |
| | Single Error in *Asana Amendment in Vīrabhadrāsana-II* | |
| 2 | *a g h il h g a* | Correctly Done Asana with right knee bend. |
| 3 | *a g h il h g a* | Asana Sequence is correct. Increase hold time of main pose with left leg. |
| 4 | *a g h il h g a* | Asana Sequence is correct. Increase hold time of main pose with right leg. |
| 5 | *a g h il h il h g a* | Left Knee straight in between. |
| 6 | *a g h ir h ir h g a* | Right Knee straight in between. |
| 7 | *a g h il g a* | Erroneous / Missing pose - Hands Apart Post-posture |
| 8 | *a g h g a* | Erroneous / Missing pose – *Vīrabhadrāsana II* main posture. |
| 9 | *g h il h g a* | *Erroneous or Missing pose - Samasthiti* in Pre-posture |
| 10 | *a g h il h g* | Erroneous / Missing pose - *Samasthiti* Missing in Post-posture |
| 11 | *a g il h g a* | Erroneous / Missing pose - Hands Apart Pre-posture |
| | Multiple Errors in *Vīrabhadrāsana II* | |
| 12 | *a g ir g a* | Erroneous / Missing pose - Hands Apart Pre-posture and Post-posture |

Table 4.19 and table 4.20 shows recognition results for correctly performed *Vīrabhadrāsana-II* with left leg and right leg respectively. Here, all seven postures are identified in correct expected sequence. *Vīrabhadrāsana -II* main posture with left leg is hold for 18 seconds and with right leg is hold for 31.5 seconds.

Most probable error while performing *Vīrabhadrāsana-II* is practitioner may not able to hold *Vīrabhadrāsana-II* main posture for sufficient time. It is expected to hold main pose for at least ten seconds / three breathings for novice practitioner. Table 4.21 is example *Vīrabhadrāsana-II* main posture not hold for sufficient time and it is hold for 4.833 seconds. As there is error in hold time and not in angle values, only abstract level amendment is provided to practitioner.

*Table 4.19: Correctly done Vīrabhadrāsana-II with left leg*

| Actual recognized sequence is : | | |
|---|---|---|
| *a* | 150 | 5.0 Sec. |
| *g* | 20 | 0.666 Sec. |
| *h* | 56 | 1.866 Sec. |
| *il* | 540 | 18.0 Sec. |
| *h* | 50 | 1.666 Sec. |
| *g* | 174 | 5.8 Sec. |
| *a* | 91 | 3.033 Sec. |
| | | |
| Correctly performed *Vīrabhadrāsana II*. Keep practicing | | |

*Table 4.20: Correctly done Vīrabhadrāsana-II with right leg*

| Actual recognized sequence is : | | |
|---|---|---|
| *a* | 160 | 5.333 Sec. |
| *g* | 54 | 1.8 Sec. |
| *h* | 55 | 1.833 Sec. |
| *ir* | 945 | 31.5 Sec. |
| *h* | 114 | 3.8 Sec. |
| *g* | 58 | 1.933 Sec. |
| *a* | 49 | 1.633 Sec. |
| | | |
| | | |
| Correctly performed *Vīrabhadrāsana II*. Keep practicing. | | |

*Table 4.21: Increase hold time of Vīrabhadrāsana-II main pose with left leg*

| Actual recognized sequence is: | | |
|---|---|---|
| *a* | 86 | 2.866 Sec. |
| *g* | 49 | 1.633 Sec. |
| *h* | 66 | 2.2 Sec. |
| *il* | 145 | 4.833 Sec. |
| *h* | 129 | 4.3 Sec. |
| *g* | 18 | 0.6 Sec. |
| *a* | 76 | 2.533 Sec. |
| | | |
| Increase Hold Time of standing on toes state to 10 Sec / 3 breathings. Current hold time is: 5.033333 sec. | | |

Most important posture in any *Yogāsana* is main posture, table 4.22 shows Erroneous / Missing pose *Vīrabhadrāsana-II* main posture.

*Table 4.22: Erroneous / Missing pose – Vīrabhadrāsana-II Main Pose*

| | | |
|---|---|---|
| Actual recognized sequence is: | | |
| *a* | 176 | 5.866 Sec. |
| *g* | 66 | 2.2 Sec. |
| *h* | 1054 | 35.133 Sec. |
| *g* | 59 | 1.966 Sec. |
| *a* | 31 | 1.033 Sec. |
| | | |
| Wrongly performed Vīrabhadrāsana II | | |
| Initially performed correct Asana postures are | | |
| *a* | Samasthiti | |
| *g* | Legs Apart | |
| After that following sequence of postures should be performed : | | |
| *h* | Legs and Hands Apart | |
| *Il* | Vīrabhadrāsana Main Pose | |
| *h* | Legs and Hands Apart | |
| *g* | Legs Apart | |
| *a* | Samasthiti | |
| | | |
| Wrongly performed Vīrabhadrāsana II | | |
| Error after frame : | 242 | 11541 |
| Error before frame : | 1185 | 12553 |

### 4.4.4. *Vṛukṣāsana*

Sample eleven cases of *Vṛukṣāsana* are discussed in table 4.23, out of that first is correct *Vṛukṣāsana* and remaining are erroneous *Vṛukṣāsana*.

Table 4.24 and table 4.25 shows recognition results for correctly performed *Vṛukṣāsana* with left leg and right leg respectively. Here, all seven postures in both variants are identified in correct expected sequence and *Vṛukṣāsana* main posture is hold for 10.166 seconds and 17.2 seconds with left and right legs respectively.

Most probable error while performing *Vṛukṣāsana* is practitioner may not able to hold *Vṛukṣāsana* main posture for at least three breathings / ten seconds i.e. for minimum required time for novice practitioner. *Yogāsana* details for main pose not hold for sufficient time are shown in table 4.26. Here, practitioner hold the required *Vṛukṣāsana* main pose for 6.2 seconds.

*Table 4.23: Possible sample sequences for Vṛukṣāsana*

| Sr. No. | Sequence | Details |
|---|---|---|
| 1 | *a il kl ml kl il a* | Correctly Done *Vṛukṣāsana* with left leg. |
| Single Error in *Asana* | | |
| 2 | *a ir kr mr kr ir a* | Standing on toes pose not hold for 10 sec./ 3 breathings |
| 3 | *a il kl ml kl* ***ml kl*** *il a* | Legs Taken Down In between along with count |
| 4 | *a ir kr mr kr* ***mr kr*** *ir a* | Legs Taken Down In between along with count |
| 5 | *a kl ml kl il a* | Erroneous / Missing pose - *Namaskār* pose in Pre-postures |
| 6 | *a kr mr kr ir a* | Erroneous / Missing pose - *Namaskār* pose in Pre-postures |
| 7 | *a il kl ml kl a* | Erroneous / Missing pose - *Namaskār* pose in Post-postures |
| 8 | *a ir kr mr kr a* | Erroneous / Missing pose - *Namaskār* pose in Post-postures |
| 9 | *a kl ml kl a* | Erroneous / Missing pose - *Namaskār* pose in Pre and Post-postures |
| 10 | *a il a* | Erroneous / Missing pose - Hands Up *Namaskār* pose |
| 11 | *a il a* | Erroneous / Missing pose - Hands Up *Namaskār* pose. |

*Table 5.24: Correctly done Vṛukṣāsana with left leg*

| Actual recognized sequence is: | | |
|---|---|---|
| *a* | 73 | 2.433 Sec. |
| *il* | 101 | 3.366 Sec. |
| *kl* | 75 | 2.5 Sec. |
| *ml* | 305 | 10.166 Sec. |
| *kl* | 138 | 4.6 Sec. |
| *il* | 104 | 3.466 Sec. |
| *a* | 80 | 2.666 Sec. |
| | | |
| *Vṛukṣāsana* performed correctly. You are doing well.... Keep it up!! | | |

*Table 5.25: Correctly done Vṛukṣāsana with Right leg*

| Actual recognized sequence is: | | |
|---|---|---|
| *a* | 142 | 4.733 Sec. |
| *ir* | 130 | 4.333 Sec. |
| *kr* | 98 | 3.266 Sec. |
| *mr* | 516 | 17.2 Sec. |
| *kr* | 83 | 2.766 Sec. |
| *Ir* | 113 | 3.766 Sec. |
| *a* | 132 | 4.4 Sec. |
| | | |
| *Vṛukṣāsana* performed correctly. You are doing well.... Keep it up!! | | |

*Table 4.26: Vṛukṣāsana main pose not hold for sufficient time*

| Actual recognized sequence is: | | |
|---|---|---|
| *a* | 142 | 4.733 Sec. |
| *ir* | 130 | 4.333 Sec. |
| *kr* | 98 | 3.266 Sec. |
| *mr* | 189 | 6.3 Sec. |
| *kr* | 83 | 2.766 Sec. |
| *ir* | 113 | 3.766 Sec. |
| *a* | 132 | 4.4 Sec. |
| | | |
| Increase Hold Time of standing on toes state to 10 Sec / 3 breathings. Current hold time is: 6.2 sec. | | |

There is possibility of erroneously performing or missing some intermediate posture in posture sequence, while performing *Vṛukṣāsana*. Table 4.27 explains case of erroneous *Namaskār* pose in pre-posture sequence.

*Table 4.27: Erroneous / Missing Posture - Namaskār pose in Pre-posture sequence*

| Actual recognized sequence is: | | |
|---|---|---|
| *a* | 150 | 5.0 Sec. |
| *il* | 55 | 1.833 Sec. |
| *ml* | 385 | 12.833 Sec. |
| *kl* | 186 | 6.2 Sec. |
| *il* | 79 | 2.633 Sec. |
| *a* | 42 | 1.4 Sec. |
| Wrongly performed *Vṛukṣāsana* | | |
| | | |
| Initially performed correct Asana postures are | | |
| *a* Samasthiti | | |
| il Knee bend | | |
| After that following sequence of postures should be performed : | | |
| ml Knee bend and *Namaskār* | | |
| Finally performed correct Asana postures are | | |
| kl *Vṛukṣāsana* main pose | | |
| ml Knee bend and *Namaskār* | | |
| il Knee bend | | |
| a Samasthiti | | |
| Error in pre-postures | | |
| | | |
| Error after frame : | 225 | 8425 |
| Error before frame : | 310 | 10550 |

Table 4.28 explains the knee bend with *Namaskār* pose missing in pre-posture and post-posture sequence. It is case of multiple errors in *Vṛukṣāsana* i.e. in pre-posture and post-posture sequence.

*Table 4.28: Knee bend with Namaskār pose missing in pre-posture and post-posture sequence*

| Actual recognized sequence is: | | |
|---|---|---|
| *a* | 166 | 5.533 Sec. |
| *il* | 114 | 3.8 Sec. |
| *ml* | 601 | 20.033 Sec. |
| *il* | 77 | 2.566 Sec. |
| *a* | 92 | 3.04 Sec. |
| Wrongly performed *Vṛukṣāsana* | | |
| | | |
| Initially performed correct *Āsana* postures are | | |
| a *Samasthiti* | | |
| il Knee bend | | |
| After that following sequence of postures should be performed : | | |
| kl Knee bend and *Namaskār* | | |
| Performed correct *Āsana* postures are | | |
| ml *Vṛukṣāsana* main pose | | |

| | | |
|---|---|---|
| After that following sequence of postures should be performed : | | |
| kl Knee bend and *Namaskār* | | |
| il Knee bend | | |
| a *Samasthiti* | | |
| Intermediate postures are not present | | |
| | | |
| Error in pre-postures | | |
| Error after frame : | 289 | 11075 |
| Error before frame : | 395 | 11181 |
| Error in post-postures | | |
| Error after frame : | 987 | 12062 |
| Error before frame : | 1121 | 12196 |

### *4.4.5. Ardhauttānāsana*

Sample eleven cases of *Ardhauttānāsana* are discussed in table 4.29, out of that first is correct *Ardhauttānāsana* and remaining are erroneous *Ardhauttānāsana*.

*Table 4.29: Possible sample sequences for Ardhauttānāsana*

| Sr. No. | Sequence | Details |
|---|---|---|
| Single Error in *Ardhauttānāsana* | | |
| 1 | $a_s$ $n_s$ $d_s$ $p_s$ $d_s$ $n_s$ $a_s$ | Correctly Done *Ardhauttānāsana* |
| 2 | $a_s$ $n_s$ $d_s$ $p_s$ $d_s$ $n_s$ $a_s$ | *Ardhauttānāsana* main pose not hold for 10 sec. |
| 3 | $a_s$ $n_s$ $d_s$ $p_s$ $d_s$ $p_s$ $d_s$ $n_s$ $a_s$ | *Ardhauttānāsana* main pose not hold constantly |
| 4 | $a_s$ $d_s$ $p_s$ $d_s$ $n_s$ $a_s$ | Erroneous / Missing pose - Hands front in Pre-postures |
| 5 | $a_s$ $d_s$ $p_s$ $d_s$ $a_s$ | Erroneous / Missing pose - Hands front in Pre-postures and Post-postures |
| 6 | $n_s$ $d_s$ $p_s$ $d_s$ $n_s$ $a_s$ | Erroneous / Missing pose - Samasthiti in Pre-postures |
| 7 | $a_s$ $n_s$ $d_s$ $p_s$ $d_s$ $n_s$ | Erroneous / Missing pose - Samasthiti in Post-postures |
| 8 | $a_s$ $n_s$ $d_s$ $n_s$ $a_s$ | Erroneous / Missing pose – *Ardhauttānāsana* |
| 9 | $a_s$ $n_s$ $p_s$ $d_s$ $n_s$ $a_s$ | Erroneous / Missing pose - Hands Up in Pre-postures |
| 10 | $a_s$ $n_s$ $d_s$ $p_s$ $n_s$ $a_s$ | Erroneous / Missing pose - Hands Up in Post-postures |
| 11 | $a_s$ $n_s$ $d_s$ $p_s$ $d_s$ $n_s$ $a_s$ $n_s$ $a_s$ $d_s$ | Erroneous / Missing pose - Extra movements are done after *Ardhauttānāsana* |

Table 4.30 shows recognition results for correctly performed *Ardhauttānāsana*. Here, all seven postures are identified in correct expected sequence and *Ardhauttānāsana* main posture is hold for 14.833 seconds.

Most probable error while performing *Ardhauttānāsana* is practitioner may not able to hold *Ardhauttānāsana* main posture for three breathings / ten seconds i.e. minimum required time for novice practitioner. *Yogāsana* details for main pose not hold for sufficient time are shown in table 4.31. Here, practitioner hold the required *Ardhauttānāsana* main pose for 0.366 seconds.

*Table 4.30: Correctly done Ardhauttānāsana*

| Actual recognized sequence is: | | |
|---|---|---|
| $a_s$ | 170 | 5.666 Sec. |
| $n_s$ | 63 | 2.1 Sec. |
| $d_s$ | 76 | 2.533 Sec. |
| $p_s$ | 445 | 14.833 Sec. |
| $d_s$ | 74 | 2.466 Sec. |
| $n_s$ | 36 | 1.2 Sec. |
| $a_s$ | 68 | 2.266 Sec. |
| *Ardhauttānāsana* performed correctly.<br>You are doing well… Keep it up!! | | |

*Table 4.31: Ardhauttānāsana main pose not hold for sufficient time*

| Actual recognized sequence is: | | |
|---|---|---|
| $a_s$ | 255 | 8.5 Sec. |
| $n_s$ | 27 | 0.9 Sec. |
| $d_s$ | 89 | 2.966 Sec. |
| $p_s$ | 11 | 0.366 Sec. |
| $d_s$ | 84 | 2.8 Sec. |
| $n_s$ | 28 | 0.933 Sec. |
| $a_s$ | 42 | 1.4 Sec. |
| Wrongly performed *Ardhauttānāsana* | | |
| | | |
| Current hold time is: 0.36667 Sec. | | |
| Increase Hold Time of *Ardhauttānāsana* state to 10 Sec | | |

*Table 4. 32: Ardhauttānāsana* main pose not hold constantly

| Actual recognized sequence is: | | |
|---|---|---|
| $a_s$ | 273 | 9.1 Sec. |
| $n_s$ | 66 | 2.2 Sec. |
| $d_s$ | 68 | 2.266 Sec. |
| $p_s$ | 163 | 5.433 Sec. |
| $d_s$ | 24 | 0.8 Sec. |
| $p_s$ | 76 | 2.533 Sec. |
| $d_s$ | 91 | 3.033 Sec. |
| $n_s$ | 98 | 3.266 Sec. |
| $a_s$ | 73 | 24.33 Sec. |
| Wrongly performed *Ardhauttānāsana* | | |
| | | |
| *Ardhauttānāsana* pose not hold constantly | | |
| Initially performed correct Asana postures are | | |
| $a_s$ Samasthiti | | |
| $n_s$ Hands front | | |
| $d_s$ Hands Up | | |
| $p_s$ *Ardhauttānāsana* | | |
| $d_s$ Hands Up | | |
| After that following sequence of postures should be performed : | | |
| $n_s$ Hands front | | |
| $a_s$ Samasthiti | | |

Next possible error is *Ardhauttānāsana* main posture not hold constantly along with count, in between pose may get disturbed and practitioner tries to adjust again with required posture. Case is discussed in table 4.32 and number of times pose disturbed is

*Table 4.33:* Missing pose - Hands front in Pre-postures and Post-postures

| Actual recognized sequence is: | | |
|---|---|---|
| $a_s$ | 94 | 3.133 Sec. |
| $d_s$ | 66 | 2.2 Sec. |
| $p_s$ | 19 | 0.633 Sec. |
| $d_s$ | 466 | 15.533 Sec. |
| $a_s$ | 36 | 1.2 Sec. |
| | | |
| Wrongly performed *Ardhauttānāsana* | | |
| | | |
| Initially performed correct Asana postures are | | |
| $a_s$ Samasthiti | | |
| After that following sequence of postures should be performed : | | |
| $n_s$ Hands front | | |
| Finally performed correct Asana postures are | | |
| $d_s$ Hands Up | | |
| $p_s$ *Ardhauttānāsana* | | |
| $d_s$ Hands Up | | |
| After that following sequence of postures should be performed : | | |
| $n_s$ Hands front | | |
| $a_s$ Samasthiti | | |
| Missing pose - Hands front in Pre-postures and Post-postures | | |

*Table 4.34: Missing pose - Ardhauttānāsana main posture*

| Actual recognized sequence is: | | |
|---|---|---|
| $a_s$ | 67 | 2.233 Sec. |
| $n_s$ | 212 | 7.066 Sec. |
| $d_s$ | 106 | 3.533 Sec. |
| $n_s$ | 232 | 7.733 Sec. |
| $a_s$ | 121 | 4.033 Sec. |
| Wrongly performed *Ardhauttānāsana* | | |
| | | |
| Initially performed correct Asana postures are | | |
| $a_s$ Samasthiti | | |
| $n_s$ Hands Front | | |
| $d_s$ Hands Up | | |
| After that following sequence of postures should be performed : | | |
| $p_s$ *Ardhauttānāsana* | | |
| $d_s$ Hands Up | | |
| Finally performed correct Asana postures are | | |
| $n_s$ Hands Front | | |
| $a_s$ Samasthiti | | |

once. There is no error in angle features, so only abstract level is represented.

### 4.4.6. *Utkaṭāsana*

Sample eleven cases of *Utkaṭāsana* are discussed in table 4.35, out of that first is correct *Utkaṭāsana* and remaining are erroneous *Utkaṭāsana*.

*Table 4.35: Possible sample sequences for Utkaṭāsana*

| Sr. No. | Sequence | Details |
|---|---|---|
| 1 | $a_a$ $n_a$ $d_a$ $o_a$ $d_a$ $n_a$ $a_a$ | Correctly Done Utkaṭāsana |
| | Single Error in *Utkaṭāsana* | |
| 2 | $a_a$ $n_a$ $d_a$ $o_a$ $d_a$ $n_a$ $a_a$ | Utkaṭāsana main pose not hold for 10 sec. |
| 3 | $a_a$ $n_a$ $d_a$ $o_a$ $d_a$ $o_a$ $d_a$ $n_a$ $a_a$ | Utkaṭāsana main pose not hold constantly |
| 4 | $a_a$ $d_a$ $o_a$ $d_a$ $n_a$ $a_a$ | Erroneous / Missing pose - Hands front in Pre-postures |
| 5 | $a_a$ $d_a$ $o_a$ $d_a$ $a_a$ | Erroneous / Missing pose - Hands front in Pre-postures and Post-postures |
| 6 | $n_a$ $d_a$ $o_a$ $d_a$ $n_a$ $a_a$ | Erroneous / Missing pose - Samasthiti in Pre-postures |
| 7 | $a_a$ $n_a$ $d_a$ $o_a$ $d_a$ $n_a$ | Erroneous / Missing pose - Samasthiti in Post-postures |
| 8 | $a_a$ $n_a$ $d_a$ $n_a$ $a_a$ | Erroneous / Missing pose - Utkaṭāsana. |
| 9 | $a_a$ $n_a$ $o_a$ $d_a$ $n_a$ $a_a$ | Erroneous / Missing pose - Hands Up in Pre-postures |
| 10 | $a_a$ $n_a$ $d_a$ $o_a$ $n_a$ $a_a$ | Erroneous / Missing pose - Hands Up in Post-postures |
| 11 | $a_a$ $n_a$ $d_a$ $o_a$ $d_a$ $n_a$ $a_a$ $n_a$ $a_a$ | Erroneous / Missing pose - Extra movements are done after Utkaṭāsana |

Table 4.36 shows recognition results for correctly performed *Utkaṭāsana*. Here, all seven postures are identified in correct expected sequence and *Utkaṭāsana* main posture is hold for 21.96 seconds.

*Table 4.36: Correctly done Utkaṭāsana*

| Actual recognized sequence is: | | |
|---|---|---|
| $a_a$ | 431 | 14.366 Sec. |
| $n_a$ | 362 | 12.066 Sec. |
| $d_a$ | 65 | 2.5 Sec. |
| $o_a$ | 343 | 11.433 Sec. |
| $d_a$ | 742 | 24.73 Sec. |
| $n_a$ | 318 | 10.6 Sec. |
| $a_a$ | 681 | 22.7 Sec. |
| Utkatasana performed correctly. You are doing well… Keep it up!! | | |

Most probable error while performing *Utkaṭāsana* is practitioner may not be able to hold *Utkaṭāsana* main posture for three breathings / ten seconds i.e. minimum required time. *Yogāsana* details for main pose not hold for sufficient time are shown in table 4.37. Here, practitioner hold the required Utkaṭāsana main pose for 0.366 seconds.

*Table 4.37: Utkaṭāsana main pose not hold for sufficient time*

| Actual recognized sequence is: | | |
|---|---|---|
| $a_a$ | 255 | 8.5 Sec. |
| $n_a$ | 27 | 0.9 Sec. |
| $d_a$ | 89 | 2.966 Sec. |
| $o_a$ | 11 | 0.366 Sec. |
| $d_a$ | 84 | 2.8 Sec. |
| $n_a$ | 28 | 0.933 Sec. |
| $a_a$ | 42 | 1.4 Sec. |
| Wrongly performed Utkatasana | | |
| Current hold time for main posture is: 0.366667<br>Increase Hold Time of Utkatasana state to 10 Sec | | |

Table 4.38 shows hands front pose missing in post-posture sequence. Practitioner have not performed hands front in post-posture sequence.

Table 4.38: Missing pose - Hands front in post-posture sequence

| Actual recognized sequence is: | | |
|---|---|---|
| $a_a$ | 70 | 2.33 Sec. |
| $n_a$ | 124 | 4.13 Sec. |
| $d_a$ | 145 | 4.83 Sec. |
| $o_a$ | 345 | 11.5 Sec. |
| $d_a$ | 123 | 4.1 Sec. |
| $a_a$ | 97 | 3.23 Sec. |
| | | |
| Wrongly performed Utkaṭāsana | | |
| Utkaṭāsana pose not hold constantly | | |
| Initially performed correct Asana postures are | | |
| $a_a$ Samasthiti | | |
| $n_a$ Hands front | | |
| $d_a$ Hands Up | | |
| $o_a$ Hands Up body Up | | |
| $d_a$ Hands Up | | |
| After that following sequence of postures should be performed: | | |
| $n_a$ Hands front | | |
| $a_a$ Samasthiti | | |
| Missing pose - Hands front in post-posture of Utkaṭāsana | | |

Three layer e-*YogaGuru* system is designed, implemented and tested for seven *Yogāsana* dataset of 1290 video sequences. e-*YogaGuru* system has successfully identified errors and suggested amendment at abstract and detail levels in erroneous *Yogāsana*. Sample tested error categories are explained in this section.

## 4.5. *Bharatnāṭyam Adavu* Recognition System from Depth Data

To study and test performance of system developed using humanoid image features a BARSDD is designed, implemented and tested. A video is captured through Kinect sensor for each *Adavu* and selected keyframes are given as an input to the system. Contours are intermediate output of recognition system and discussed in chapter 3. The system is trained for five *Adavu* and tested for five different dancers. System recognizes postures from dance sequence of *Adavu* and failure of recognition of complete *Adavu* is due to non-recognition of intermediate states. Average recognition rate is 78.38%. Recognition details are explained in Table 4.39.

*Table 4.39: Recognition results for Adavu*

| Test No. / Adavu | Test1 (%) | Test 2 (%) | Test 3 (%) | Test 4 (%) | Test 5 (%) | Avg % |
|---|---|---|---|---|---|---|
| **Tatta** | 100 | 80 | 80 | 60 | 80 | 80 |
| **Natta** | 83.30 | 83.30 | 83.30 | 66.66 | 83.30 | 79.90 |
| **Sarikal** | 87.5 | 87.5 | 87.5 | 75 | 75 | 79.41 |
| **Visharu** | 74.07 | 74.07 | 74.07 | 70.37 | 70.37 | 72.59 |
| **Kudittametta** | 82.35 | 82.35 | 82.35 | 76.47 | 76.47 | 80 |

Kinematic model is finalized for further experimentation of recognition and suggestion of amendment due to its good recognition performance as compared to image model. In kinematic model angle features found to be more discriminative, robust and less variant to anthropometry. Chapter first explained about intermediate features and results, then results of recognition of *Yogāsana* using simple state model and joint features. Further, results of recognition of *Yogāsana* using angle features, it is observed that system gave almost 100% results. The results of suggestion of amendment are tested on seven *Yogāsana*. Results of *Bharatnāṭyam Adavu* recognition system using image feature are discussed in last section and system gave 78.38% average recognition rate. It is very poor as compared to *Yogāsana* recognition using kinematic features.

Chapter 5

# CONCLUSION
# AND
# FUTURE SCOPE

# 5. CONCLUSION AND FUTURE SCOPE

*"In literature and in life we ultimately pursue, not conclusions, but beginnings."*

*— Sam Tanenhaus, Literature Unbound*

System named e-*YogaGuru* designed and developed to assess the correctness of performed *Yogāsana*, and to provide corrective suggestion in erroneous *Yogāsana* to provide feedback to the practitioner. The conclusions of all the experiments carried out during this research work are summarized in this chapter.

## 5.1. Conclusions

From exhaustive literature survey it is observed that very few researchers worked on suggestion of amendment aspect of human motion recognition domain. Even though performing *Yogāsana* has tremendous benefits, standard dataset is not available for it. In this research work system *e-YogaGuru* for suggestion of amendment in human action is designed, developed and tested. Major contributions of this research are summarized below-

- System implemented using designed algorithms provides assistance for a person to perform *Yogāsana* at home in absence of expert. System also provides flexibility of time and comfort to perform *Yogāsana*.
- Detailed amendment in erroneous *Yogāsana* at incorrect posture in terms of body angle is suggested in detail and it helps practitioner to improve for next iteration. Stored *Yogāsana* data and replay facility helps trainer to analyze movements and provide more expert suggestions to practitioner.
- State model designed and implemented in *e-YogaGuru* system provides advantage of execution speed invariance and reduced size of a knowledge base. Use of state model also helps to analyze the *Yogāsana* for suggestion of amendment, where other recognition approached may fail.
- Use of human body angle features for representation of motion provides advantage of more accurate and discriminative features, reduced feature vector and anthropometric invariance.

- Created dataset of 1290 video sequences of *Yogāsana* with approximately 23, 22,000 frames for seven *Yogāsana* can be utilized by other researchers for testing their ideas and algorithms. *Bharatnāṭyam* dataset of depth data for five *Adavu* is also created.

- Feature selection is most important step in human motion recognition. Two broad options available for human body representation are using humanoid body model features and humanoid image model features. For performance analysis of features designed, developed and tested *Bharatnāṭyam Adavu* recognition system using human body contour (humanoid image) features for five *Adavu* and *Yogāsana* recognition system using human body joint positions (humanoid body) as feature. From experimentation it is observed that system with human contours as features is very sensitive to human anthropometry and clothing as compared to system with joint positions. Recognition rate of system with kinematic features is high. So, use of kinematic features strongly recommended for suggestion of amendment.

## 5.2. Future Scope

There is always scope for improvement of any system. There are many challenges in the domain of human motion recognition and this will definitely keep researchers active to propose more robust and accurate solutions. Following are some of the identified future scopes -

- Online version of designed system will be more beneficial for monitoring the performance of practitioner. Also, systems will provide online expert's opinion on performed movement and this can be added as incremental learning to system.

- In future, model designed for *e-YogaGuru* system can be applied for other similar applications like different exercises, dances etc. System can be used for recognition of other *Yogāsana* by modifying knowledge base.

- Recognition of *Adavu* in *Bharatnāṭyam* dance is basic step for recognition of story. System can be modified to recognize *Bharatnāṭyam* story from dancer's facial sentiments and body posture. Same should be displayed for audience in textual form.

# PUBLICATIONS

## Patent

- Indian patent filed- **Patent No:** 201621029482 **Date:** 30/08/2016 **Title:** System and method for motion analysis and feedback for amendment of human action.

## Publications

- G. V. Kale, V. H. Patil, A Study of Vision based Human Motion Recognition and Analysis, *International Journal of Ambient Computing and Intelligence IGI Global*, vol. 7, no. 2, pp. 75-92, 2016.

- G. V. Kale, V. H. Patil, *Bharatnāṭyam Adavu* Recognition from Depth Data, in *Image Information Processing (ICIIP), 2015 Third International Conference on*, Solan, 2015, pp. 246-251.

- G. V. Kale, V. H. Patil, "Real Time Human Action Recognition using Kinematic State Model", *Advances in Image and Video Processing*, vol. 2, no. 6, pp. 17-22, 2014.

- G. V. Kale, V. H. Patil, Motion correction in physical action of Human body applied to *Yogāsana*, *Advances in Computing and Information Technology (ACITY)*, 2012 *Second International Conference on,* Chennai, India, 2016, vol.2, pp. 675-680.

- G. V. Kale, Survey of vision based representation and recognition methods, cPGCON-14, Research Scholar forum, Savitribai Phule Pune University.

- G. V. Kale, Presentation poster paper in Innovation-15, state level conference by SPPU BCUD, titled, Vision based human motion recognition and analysis in indoor scene.

- G. V. Kale, Vision based human motion recognition and analysis in indoor scene, Innovation-16, state level conference by SPPU, Pune.

- G. V. Kale, Presented paper at 11th Inter-Research-Institute Student Seminar in Computer Science (IRISS 2017) at Kolkata- A Study of Vision based Human Motion Recognition and Analysis, *International Journal of Ambient Computing and Intelligence*, vol. 7, no. 2, pp. 75-92, 2016.

## Funded Project

Successfully utilized funds received from SPPU BCUD amount Rs. 1.9 lacks for research work. **Letter No.** 14ENG002310, **Duration:** 2014-16 for, “Vision based human motion recognition and analysis in indoor scene for activity monitoring using distributed data capture”, which is part of PhD work.

# REFERENCES

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

2. Aggarwal, J.K., Cai, Q., Liao, W. and Sabata, B., 1998. Nonrigid motion analysis: Articulated and elastic motion. *Computer Vision and Image Understanding*, *70*(2), pp.142-156.

3. Aggarwal, J.K. and Ryoo, M.S., 2011. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, *43*(3), p.16.

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

5. Poppe, R., 2007. Vision-based human motion analysis: An overview. *Computer vision and image understanding*, *108*(1), pp.4-18.

6. Turaga, P., Chellappa, R., Subrahmanian, V.S. and Udrea, O., 2008. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, *18*(11), pp.1473-1488.

7. Han, J., Shao, L., Xu, D. and Shotton, J., 2013. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE transactions on cybernetics*, *43*(5), pp.1318-1334.

8. Makris, D. and Ellis, T., 2005. Learning semantic scene models from observing activity in visual surveillance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *35*(3), pp.397-408.

9. Morris, B.T. and Trivedi, M.M., 2008. A survey of vision-based trajectory learning and analysis for surveillance. *IEEE transactions on circuits and systems for video technology*, *18*(8), pp.1114-1127.

10. Neves, J., Narducci, F., Barra, S. and Proença, H., 2016. Biometric recognition in surveillance scenarios: a survey. *Artificial Intelligence Review*, *46*(4), pp.515-541.

11. Alam, M.R., Reaz, M.B.I. and Ali, M.A.M., 2012. A review of smart homes—Past, present, and future. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(6), pp.1190-1203.

12. Sarkar, S., Phillips, P.J., Liu, Z., Vega, I.R., Grother, P. and Bowyer, K.W., 2005. The humanid gait challenge problem: Data sets, performance, and analysis. *IEEE transactions on pattern analysis and machine intelligence*, *27*(2), pp.162-177.

13. Sepehri, A., Yacoob, Y. and Davis, L.S., 2006. Employing the hand as an interface device. *Journal of Multimedia*, *1*(7), pp.18-29.

14. Ronchetti, M. and Avancini, M., 2011. Using kinect to emulate an interactive whiteboard. *MS in Computer Science*, *University of Trento*.

15. Hoffmann, M., Marques, H., Arieta, A., Sumioka, H., Lungarella, M. and Pfeifer, R., 2010. Body schema in robotics: a review. *IEEE Transactions on Autonomous Mental Development*, *2*(4), pp.304-324.

16. Leo, M., Medioni, G., Trivedi, M., Kanade, T. and Farinella, G.M., 2017. Computer vision for assistive technologies. *Computer Vision and Image Understanding*, *154*, pp.1-15.

17. Najafi, B., Aminian, K., Paraschiv-Ionescu, A., Loew, F., Bula, C.J. and Robert, P., 2003. Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly. *IEEE Transactions on biomedical Engineering*, *50*(6), pp.711-723.

18. Lin, J.F.S. and Kulic, D., 2014. Online segmentation of human motion for automated rehabilitation exercise analysis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *22*(1), pp.168-180.

19. Bertini, M., Del Bimbo, A. and Nunziati, W., 2003, November. Model checking for detection of sport highlights. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval* (pp. 215-222). ACM.

20. Watanabe, T., Shibusawa, S., Kamada, M., Yonekura, T. and Ohtsuka, N., 2015. Design and Development of Lower Limb Chair Exercise Support System with Depth Sensor. *Transactions on Networks and Communications*, *3*(4), p.30.

21. Dao, T.T., Tannous, H., Pouletaut, P., Gamet, D., Istrate, D. and Tho, M.H.B., 2016. Interactive and connected rehabilitation systems for E-Health. *Irbm*, *37*(5), pp.289-296.

22. Kale, G. and Patil, V., 2015, December. Bharatnāṭyam Adavu Recognition from Depth Data. In *Image Information Processing (ICIIP), 2015 Third International Conference on* (pp. 246-251). IEEE.

23. Field, T., 2011. Yoga clinical research review. *Complementary therapies in clinical practice*, *17*(1), pp.1-8.

24. Kauts, A. and Sharma, N., 2009. Effect of yoga on academic performance in relation to stress. *International journal of yoga*, *2*(1), p.39.

25. Williams, K.A., Petronis, J., Smith, D., Goodrich, D., Wu, J., Ravi, N., Doyle, E.J., Juckett, R.G., Kolar, M.M., Gross, R. and Steinberg, L., 2005. Effect of Iyengar yoga therapy for chronic low back pain. *Pain*, *115*(1), pp.107-117.

26. Moeslund, T.B., Hilton, A. and Krüger, V., 2006. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, *104*(2), pp.90-126.

27. Moeslund, T.B. and Granum, E., 2001. A survey of computer vision-based human motion capture. *Computer vision and image understanding*, *81*(3), pp.231-268.

28. Cedras, C. and Shah, M., 1995. Motion-based recognition a survey. *Image and Vision Computing*, *13*(2), pp.129-155.

29. Wang, L., Hu, W. and Tan, T., 2003. Recent developments in human motion analysis. *Pattern recognition*, *36*(3), pp.585-601.

30. Marinoiu, E., Papava, D. and Sminchisescu, C., 2013. Pictorial human spaces: How well do humans perceive a 3d articulated pose?. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1289-1296).

31. Han, J., Shao, L., Xu, D. and Shotton, J., 2013. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE transactions on cybernetics*, *43*(5), pp.1318-1334.

32. Khoshelham, K. and Elberink, S.O., 2012. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, *12*(2), pp.1437-1454.

33. Obdržálek, Š., Kurillo, G., Ofli, F., Bajcsy, R., Seto, E., Jimison, H. and Pavel, M., 2012, August. Accuracy and robustness of Kinect pose estimation in the context of coaching of elderly population. In *Engineering in medicine and biology society (EMBC), 2012 annual international conference of the IEEE* (pp. 1188-1193). IEEE.

34. Galna, B., Barry, G., Jackson, D., Mhiripiri, D., Olivier, P. and Rochester, L., 2014. Accuracy of the Microsoft Kinect sensor for measuring movement in people with Parkinson's disease. *Gait & posture*, *39*(4), pp.1062-1068.

35. Smisek, J., Jancosek, M. and Pajdla, T., 2013. 3D with Kinect. In *Consumer depth cameras for computer vision* (pp. 3-25). Springer London.

36. Stoyanov, T., Louloudi, A., Andreasson, H. and Lilienthal, A.J., 2011. Comparative evaluation of range sensor accuracy in indoor environments. In *5th European Conference on Mobile Robots, ECMR 2011, September 7-9, 2011, Örebro, Sweden* (pp. 19-24).

37. Gao, Z., Song, J.M., Zhang, H., Liu, A.A., Xue, Y.B. and Xu, G.P., 2014. Human action recognition via multi-modality information. *Journal of Electrical Engineering and Technology*, *9*(2), pp.739-748.

38. Yang, X. and Tian, Y., 2014. Effective 3d action recognition using eigenjoints. *Journal of Visual Communication and Image Representation*, *25*(1), pp.2-11.

39. Wang, Q., Kurillo, G., Ofli, F. and Bajcsy, R., 2015, October. Evaluation of pose tracking accuracy in the first and second generations of microsoft kinect. In *Healthcare Informatics (ICHI), 2015 International Conference on* (pp. 380-389). IEEE.

40. Rao, C. and Shah, M., 2001. View-invariance in action recognition. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* (Vol. 2, pp. II-II). IEEE.

41. Johansson, G., 1973. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, *14*(2), pp.201-211.

42. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M. and Moore, R., 2013. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, *56*(1), pp.116-124.

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

44. Sminchisescu, C. and Triggs, B., 2003. Estimating articulated human motion with covariance scaled sampling. *The International Journal of Robotics Research*, *22*(6), pp.371-391.

45. Yamato, J., Ohya, J. and Ishii, K., 1992, June. Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on* (pp. 379-385). IEEE.

46. Bobick, A.F., 1997. Movement, activity and action: the role of knowledge in the perception of motion. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, *352*(1358), pp.1257-1265.

47. Bobick, A.F. and Davis, J.W., 2001. The recognition of human movement using temporal templates. *IEEE Transactions on pattern analysis and machine intelligence*, *23*(3), pp.257-267.

48. Sheikh, Y., Sheikh, M. and Shah, M., 2005, October. Exploring the space of a human action. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (Vol. 1, pp. 144-149). IEEE.

49. Campbell, L.W. and Bobick, A.F., 1995, June. Recognition of human body motion using phase space constraints. In *Computer Vision, 1995. Proceedings., Fifth International Conference on* (pp. 624-630). IEEE.

50. Wu, D. and Shao, L., 2013. Silhouette analysis-based action recognition via exploiting human poses. *IEEE Transactions on Circuits and Systems for Video Technology*, *23*(2), pp.236-243.

51. Shechtman, E. and Irani, M., 2005, June. Space-time behavior based correlation. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (Vol. 1, pp. 405-412). IEEE.

52. Laptev, I. and Lindeberg, T., 2004, August. Velocity adaptation of space-time interest points. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on* (Vol. 1, pp. 52-56). IEEE.

53. Veeraraghavan, A., Chellappa, R. and Roy-Chowdhury, A.K., 2006. The function space of an activity. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (Vol. 1, pp. 959-968). IEEE

54. Su, C.J., 2013. Personal rehabilitation exercise assistant with kinect and dynamic time warping. *International Journal of Information and Education Technology*, *3*(4), p.448.

55. Oliver, N.M., Rosario, B. and Pentland, A.P., 2000. A Bayesian computer vision system for modeling human interactions. *IEEE transactions on pattern analysis and machine intelligence*, *22*(8), pp.831-843.

56. Mansur, A., Makihara, Y. and Yagi, Y., 2013. Inverse dynamics for action recognition. *IEEE transactions on cybernetics*, *43*(4), pp.1226-1236.

57. M. Yang, 2014. “Human Action Recognition Based on Kinect”, Journal of Computational Information Systems 10(12), pp 5347–5354.

58. Nowozin, S. and Shotton, J., 2012. Action points: A representation for low-latency online human action recognition. *Microsoft Research Cambridge, Tech. Rep. MSR-TR-2012-68*.

59. Sung, J., Ponce, C., Selman, B. and Saxena, A., 2011. Human Activity Detection from RGBD Images. *plan, activity, and intent recognition*, *64*.

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

61. Ryoo, M.S. and Aggarwal, J.K., 2009. Semantic representation and recognition of continued and recursive human activities. *International journal of computer vision*, *82*(1), pp.1-24.

62. Yao, L., Xu, H. and Li, A., 2014. Kinect-based rehabilitation exercises system: therapist involved approach. *Bio-medical materials and engineering*, *24*(6), pp.2611-2618.

63. Chen, H.T., He, Y.Z., Hsu, C.C., Chou, C.L., Lee, S.Y. and Lin, B.S.P., 2014, January. Yoga Posture Recognition for Self-training. In *International Conference on Multimedia Modeling* (pp. 496-505). Springer International Publishing.

64. Zimmerman, J., Zimmerman Jeffrey C., 2003. *Systems and methods for motion analysis and feedback*. U.S. Patent Application 10/358,396.

65. Chaquet, J.M., Carmona, E.J. and Fernández-Caballero, A., 2013. A survey of video datasets for human action and activity recognition. *Computer Vision and Image Understanding*, *117*(6), pp.633-659.

66. Pham, C.H., Le, Q.K. and Le, T.H., 2014. Human action recognition using dynamic time warping and voting algorithm. *VNU Journal of Science: Computer Science and Communication Engineering*, *30*(3).

67. Iyengar, B.K.S., 2011. *The illustrated light on yoga*. Orient Black Swan.