SEGMENTATION OF WUSHU KICKING MOTION FROM MOTION CAPTURE DATA USING MATLAB



Compiled By:

N a m a : M Farrel R Fajra
NIM : 21523136

INFORMATICS STUDY PROGRAM - UNDERGRADUATE PROGRAM
FACULTY OF INDUSTRIAL TECHNOLOGY
ISLAMIC UNIVERSITY INDONESIA
2025

ENDORSEMENT PAGE SUPERVISOR

MODELING OF MARTIAL ARTS MOVEMENTS USING MOTION CAPTURE

THESIS

Compiled By:

N a m e : M Farrel R Fajra

NIM : 21523136

Yogyakarta, July 23rd, 2025

Mentor,

23/07/2025

(Izzati Muhimmah, S.T., M.Sc., Ph.D.)

RATIFICATION PAGE OF THE EXAMINING LECTURER

SEGMENTATION OF WUSHU KICKING MOTION FROM MOTION CAPTURE DATA USING MATLAB

FINAL TASK

Has been defended in front of the examiner as one of the requirements to obtain a Bachelor of Computer Science degree from the Informatics Study Program - Undergraduate Program

at the Faculty of Industrial Technology, Universitas Islam Indonesia

Yogyakarta, August 4, 2025

Testing Team

Izzati Muhimmah, S.T., M.Sc., Ph.D.

Member 1

Sheila Nurul Huda, S.Kom., M.Cs.

Member 2

Mukhammad Andri Setiawan, S.T., M.Sc., Ph.D.

Knowing,

Head of Informatics Study Program - Undergraduate Program

Faculty of Industrial Technology

Islamic University of Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

FINAL PROJECT AUTHENTICITY STATEMENT PAGE

The undersigned:

Name : M Farrel R Fajra

NIM : 21523136

Final project with title:

MODELING OF MARTIAL ARTS MOVEMENTS USING MOTION CAPTURE

Stating that all components and contents in this final project are my own work. If in the future it is proven that some parts of this work are not my own work, the final project submitted as my own work is ready to be withdrawn and ready to bear any risks and consequences.

Thus this statement letter is made, hopefully it can be used properly.

Yogyakarta, July 23rd, 2025



(M Farrel R Fajra)

DEDICATION PAGE

Alhamdulillahirabbil 'alamin, I extend my deepest gratitude to Allah *subhanahu wa ta'ala*, for His blessings and grace, which have enabled me to complete this final project in good health and well-being. Peace and blessings be upon the Prophet Muhammad, *shalallahu 'alaihi wasallam*, the timeless role model, along with his family, companions, and all of his followers until the end of time.

First and foremost, I would like to express my deepest thanks to everyone who has stood by me from the beginning of my life up to this point. I would like to dedicate this work to:

Allah SWT, who has bestowed upon me countless blessings, opportunities, good health, and all of His goodness, allowing me to stand strong, healthy, and full of gratitude.

My beloved and respected family — my father Ruslan Husin and mother Desfitriani, sister Tisya, brothers Dimo and Rieza, and my younger sister Ai — who have always been a place I can return to. Through your endless support, guidance, prayers, and encouragement, I was able to complete this final project and my studies at the Islamic University of Indonesia to the fullest. I hope I can make you proud.

My esteemed supervisor, Dr. Izzati Muhimmah, who has kindly given her time to guide me, entrusted me with this research topic, and provided me with a great opportunity to execute and complete it.

My partner, Solehah Solahuddin, who has been my place of refuge and the one I confide in, and who, with patience, prayers, and love, has continued to strengthen me. Thank you for always being by my side, listening, and understanding me throughout this journey, allowing me to pursue my studies with enthusiasm.

My fellow friends whom I deeply respect and cherish, who have supported, motivated, accompanied, entertained, and filled the void during my academic years. I hope we stay connected into old age.

MOTO PAGE

"All I can say is don't fall at the last fence. The finishing post's in sight. See you in the paddock… keep your eye on the ball."

—*Eugene Helpmann,* from *Brazil (1985)*

FOREWORD

I thank Allah SWT for His guidance, strength, and blessings that helped me complete this research. This paper, titled "Segmentation of Wushu Kicking Motion From Motion Capture Data Using MATLAB," was written to fulfill the graduation requirement for the Informatics Study Program, Faculty of Industrial Technology, Islamic University of Indonesia.

This paper reflects the result of my effort and dedication. Even through the many challenges I faced, I was fortunate to have the support and encouragement of people around me, which allowed me to complete this work as best as I could.

I want to show my deepest appreciation to:

1. Allah SWT, for always providing a way forward even in the hardest moments.
2. My dearest family, who gave endless support, prayers, and strength. You are my foundation.
3. Dr. Raden Teduh Dirgahayu, S.T., M.Sc., as Head of the Informatics Department, Faculty of Industrial Technology, Islamic University of Indonesia.
4. Dr. Dhomas Hatta Fudholi, S.T., M.Eng., as Head of the Undergraduate Informatics Study Program, Faculty of Industrial Technology, Islamic University of Indonesia.
5. Ms. Izzati Muhimmah, S.T., M.Sc., Ph.D., as my supervisor, who guided me with patience, valuable insight, and consistent encouragement from start to finish.
6. Solehah Solahuddin, for always being there with kindness, support, and understanding through everything.
7. All my dear friends, who supported, motivated, and kept me going throughout this journey.

I hope this paper can be useful for future research in motion analysis, martial arts digitization, or other related fields in computer science and biomechanics.

Yogyakarta, July 23rd, 2025



(M Farrel R Fajra)

SARI

Motion segmentation plays a crucial role in analyzing complex martial arts movements, such as Wushu kicking, which often involve non-repetitive, high-variation actions. Traditional motion segmentation methods struggle to effectively capture transitions in such irregular movements. This study adapts an existing motion segmentation framework by Sonoda et al. (2008), using three key features—angular velocity, direction change, and volume change—to segment Wushu MoCap data. The research implements these methods on 3D motion capture data of Wushu kicks, with preprocessing steps carried out in MATLAB for visualization and analysis. The performance of each segmentation method is evaluated by comparing the results against ground truth data collected from experienced martial artists. Results show that angular velocity performs best, achieving a similarity score of 0.9574 and correctly matching 11 of 12 ground truth transitions. Direction change and volume change both score 0.8660, but each matches a different subset of transitions, highlighting the unique strengths and limitations of each approach. The integrated method, while offering broader coverage, did not improve over angular velocity alone, suggesting that simple fusion may propagate errors from individual methods. This work confirms that rotational motion is a reliable cue for detecting phase transitions in martial arts. The study's limitations include its reliance on a single Wushu kicking sequence, the challenges of capturing subtle transitions using certain methods, and the fixed 100-frame tolerance window. Future work should focus on dynamically adjusting the tolerance window, refining threshold computation to mitigate jitter effects, expanding the dataset to include a wider variety of martial arts techniques, incorporating upper body motions, and exploring real-time applications in coaching systems. Additionally, comparisons with machine learning-based segmentation methods could further enhance the robustness and accuracy of the framework.

Keywords: martial arts, motion capture, motion segmentation, angular velocity, digital curvature, volume difference, MATLAB.

GLOSSARY

MoCap	A technology that records the movement of objects or people, often using optical tracking systems.
Segmentation	The process of dividing continuous motion into discrete phases or meaningful units.
Angular Velocity	The rate of change of joint angles over time, representing rotational motion.
Digital Curvature	The rate of change in direction over time in a marker's trajectory.
Convex Hull	The smallest convex polygon or polyhedron that encloses a set of points (body markers in this case).
MATLAB	A high-level programming language and environment used for data analysis, visualization, and numerical computation
Frame	A single time point in motion capture data, typically captured at 100 frames per second.
Ground Truth	Manually annotated reference data used to validate the accuracy of automated segmentation.
Cosine Similarity	A metric used to measure the similarity between two sets of data. It compares the predicted segmentation points with ground truth points by considering the alignment between them.
Threshold	A predefined value used to filter out insignificant fluctuations in motion data, ensuring that only meaningful motion transitions are captured.

TABLE OF CONTENTS

TITLE PAGE i
ENDORSEMENT PAGE SUPERVISOR ii
RATIFICATION PAGE OF THE EXAMINING LECTURER iii
FINAL PROJECT AUTHENTICITY STATEMENT PAGE iv
DEDICATION PAGE v
MOTO PAGE vi
FOREWORD vii
SARI viii
GLOSSARY ix
TABLE OF CONTENTS x
LIST OF TABLES xii
FIGURE LIST xiii
CHAPTER I INTRODUCTION 1
1.1 Study Background 1
1.2 Problem Statement and Research Questions 2
1.3 Research Scope 2
1.4 Research Objectives 3
1.5 Report Structure 3
CHAPTER II THEORETICAL FRAMEWORK & LITERATURE REVIEW 4
2.1 Theoretical Framework 4
2.1.1 Martial Arts 4
2.1.2 Motion Capture 5
2.1.3 Motion Segmentation 6
2.1.4 Ground Truth as an Evaluation Metric 9
2.2 Literature Review 10
CHAPTER III RESEARCH METHODOLOGY 21
3.1 Preprocessing 22
3.1.1 Collecting and Preprocessing Motion Capture Data 22
3.1.2 Processing Data and Visualizing in MATLAB 26
3.2 Segmentation 29
3.2.1 Segmentation by Angular Velocity 30
3.2.2 Segmentation by Change of Direction 36
3.2.3 Segmentation by Change of Volume 39
3.2.4 Integrating Segmentation Method Results 41
3.3 Evaluation Against Ground Truth Data 45
CHAPTER IV RESULTS AND DISCUSSIONS 48
4.1 Results 48
4.1.1 Segmentation by Angular Velocity 48
4.1.2 Segmentation by Change of Direction 53
4.1.3 Segmentation by Change of Volume 58
4.1.4 Integrating Segmentation Method Results 62
4.1.5 Evaluation Against Ground Truth Data 69
4.2 Discussions 73
4.2.1 Performance Comparison of Segmentation Methods 73
4.2.2 Effectiveness of the Integrated Method 74
4.2.3 Interpretation of Findings 75
4.2.4 Comparison with Related Works 75

4.2.5 Limitations and Future Works 76
4.2.6 Summary of Findings 77
CHAPTER V CONCLUSION 80
5.1 Conclusion 80
5.2 Suggestions 81
LITERATURE 83
APPENDIX 86

LIST OF TABLES

Table 2.1 Literature review from relevant papers.....11
Table 3.1 Wushu kicking motion coordinates for marker ARIEL from frame 1 to 9 23
Table 3.2 Wushu kicking motion coordinates after VBA code.....25
Table 3.3 Wushu kicking motion coordinates after concatenation.....26
Table 4.1 Skeleton frame candidates from each three method after postprocessing.....64
Table 4.2 Result of evaluation metrics 72

FIGURE LIST

Figure 3.1 Research methodology flowchart 22
Figure 3.2 Full-body marker position and labels 23
Figure 3.3 VBA code for adding new columns automatically 24
Figure 3.4 Excel formula used for concatenation 25
Figure 3.5 MATLAB code snippet to parse coordinates 28
Figure 3.6 Skeleton visualization 29
Figure 3.7 Leg structure comparison 32
Figure 3.8 MATLAB code snippet to compute joint angles 33
Figure 3.9 MATLAB code snippet to compute angular velocity 33
Figure 3.10 MATLAB code snippet to sum and square angular velocity 34
Figure 3.11 MATLAB code snippet for Equation (4) 34
Figure 3.12 MATLAB code snippet for angular velocity's segmentation 35
Figure 3.13 MATLAB code snippet for Equation (5) and (6) 37
Figure 3.14 MATLAB code snippet for Equation (7) 37
Figure 3.15 MATLAB code snippet for Equation (8) 38
Figure 3.16 MATLAB code snippet for digital curvature's segmentation 39
Figure 3.17 MATLAB code snippet for Equation (9) 40
Figure 3.18 MATLAB code snippet for volume diff segmentation and Equation (10) 41
Figure 3.19 MATLAB code snippet for preprocessing Method 1 42
Figure 3.20 MATLAB code snippet for preprocessing Method 2 43
Figure 3.21 MATLAB code snippet for preprocessing Method 3 44
Figure 3.22 MATLAB code snippet to compute evaluation 47
Figure 4.1 Plot of angular velocity for each frame 49
Figure 4.2 Plot of angular velocity with segmentation candidates 49
Figure 4.3 19 segmented frames according to angular velocity 53
Figure 4.4 Plot of digital curvature for each frame 54
Figure 4.5 Plot of digital curvature with segmentation candidates 54
Figure 4.6 Plot of Wushu kicking motion trajectory in an XY-horizontal plane 55
Figure 4.7 Plot of Wushu kicking motion trajectory with segmentation candidates 55
Figure 4.8 16 segmented frames according to directional change in digital curvature 58
Figure 4.9 Plot of volume difference for each frame 59
Figure 4.10 Plot of volume difference with segmentation candidates 59
Figure 4.11 16 segmented frames according to volume change 62

Figure 4.12 Plot of angular velocity after postprocessing 63
Figure 4.13 Plot of digital curvature after postprocessing 63
Figure 4.14 Plot of volume difference after postprocessing 64
Figure 4.15 Raster plot showing the integrated and postprocessed result 68
Figure 4.16 12 segmented frames according to integration 69
Figure 4.17 12 segmented frames according to ground truth data 71
Figure 4.18 Raster plot comparing 4 methods against ground truth data 72
Figure 4.19 Evaluation results visualized in grouped bar chart 73

CHAPTER I
INTRODUCTION

1.1 Study Background

Martial arts are deeply intertwined with world history, serving a multitude of purposes and values. As a form of combat, martial arts not only teach practitioners physical skills such as self-defense, fitness, and motor coordination but also cultivate mental benefits, including self-esteem, respect, and discipline (Ciaccioni et al., 2024; Stamenković et al., 2022). Furthermore, martial arts play a crucial role in shaping etiquette. Research on the Indonesian martial art Pencak Silat suggests that it helps develop ethical values essential for cultivating good manners, behavior, and leadership (Schiff, 2022). In addition, martial arts often incorporate religious and philosophical elements. A study on the Chinese martial art Wushu indicates that its artistry and techniques integrate the philosophical and religious values of Buddhism, Taoism, and the concept of yin and yang (Han et al., 2021). Moreover, due to their structured movement systems and emphasis on motor skill learning, martial arts have also been recognized as a valuable domain for modelling psychomotor performance through digital techniques (Echeverria & Santos, 2021).

Recent advancements in technology have created new opportunities for the study and visualization of martial arts movements. Among these innovations, motion capture technology has emerged as a rapidly expanding field due to its precision in recording human motion and its broad range of applications, despite its inherent complexity (Wan Idris et al., 2019). Wearable sensors, as reviewed in recent studies, have expanded motion analysis applications into healthcare, rehabilitation, and intelligent motion control, further enhancing the versatility of motion capture systems (Wang et al., 2023). Beyond industrial use, motion capture has gained widespread adoption in areas such as sports performance analysis, gaming, and biomechanics, where the ability to accurately record and replicate intricate human movements is crucial. In the domain of martial arts, datasets capturing expert and novice karate techniques using optical MoCap further emphasize its role in performance benchmarking and skill differentiation (Szczęsna et al., 2021). In 2023, the global 3D motion capture system market was valued at approximately USD 222.65 million, highlighting the growing demand for this technology across multiple sectors (SNS Insider, 2023).

Motion segmentation, the process of dividing continuous movement into meaningful segments, is a key step in motion analysis. Many existing methods are designed for highly periodic or cyclic motions such as walking or running. While martial arts kicks may appear repetitive, their transitions often vary in timing and intensity, making segmentation more challenging. Compared to fields like dance, where segmentation has been more widely studied, research on segmenting martial arts movements using motion capture data remains limited.

This research adapts a segmentation framework from Sonoda et al., which identifies movement transitions based on three features: **angular velocity**, **direction change**, and **volume change**. By modeling martial arts movement through segmentation and visualization, this study aims to support deeper understanding in the martial arts domain and provide a segmentation framework that supports studies in biomechanics.

1.2 Problem Statement and Research Questions

Based on the study background, the problems this research is trying to solve are:

1. RQ1: How can Sonoda et al.'s segmentation methods be adapted to effectively detect motion phases in martial arts movements?
2. RQ2: Among angular velocity, direction change, and volume change, which method provides the most accurate segmentation?
3. RQ3: How does combining the three segmentation methods affect performance compared to using each method individually?
4. RQ4: How do the segmentation results compare with expert-annotated ground truth data?

1.3 Research Scope

This study uses 3D motion capture data of Wushu kicking movements from a publicly available dataset. It applies three segmentation methods: angular velocity, direction change, and volume change. The angular velocity method focuses on lower-body markers to analyze leg movements, while the direction change method uses a single marker to represent the body's core. The volume change method examines overall body structure using all available markers.

The analysis is limited to one data input of Wushu kicking motions and evaluated against expert-annotated ground truth. Other martial arts styles, machine learning-based segmentation, and upper-body-specific techniques are outside the scope of this research.

1.4 Research Objectives

The main objectives of this research are to:

1. Adapt a motion segmentation framework from previous work for use in the context of martial arts.
2. Analyze the effectiveness of three segmentation methods (angular velocity, direction change, and volume change) when applied to martial arts, specifically Wushu kicking motions.
3. Compare the performance of each method and evaluate their accuracy using ground truth annotations, measured by cosine similarity metric.
4. Provide an adapted segmentation framework that supports motion analysis and future research on martial arts motion data.

1.5 Report Structure

This thesis is organized as follows:

CHAPTER I INTRODUCTION: Presents the background, research questions, objectives, scope, and structure of the study.

CHAPTER II THEORETICAL FRAMEWORK AND LITERATURE REVIEW: Covers key concepts and related studies on martial arts, motion capture, and segmentation techniques.

CHAPTER III RESEARCH METHODOLOGY: Explains the dataset, preprocessing steps, segmentation methods, and evaluation metrics used in the study.

CHAPTER IV RESULTS AND DISCUSSIONS: Presents the results of each method, evaluates their performance, and compares the findings with related work.

CHAPTER V CONCLUSION: Summarizes the main findings, highlights contributions and limitations, and offers directions for future research.

CHAPTER II
THEORETICAL FRAMEWORK & LITERATURE REVIEW

2.1 Theoretical Framework

2.1.1 Martial Arts

Martial arts refer to a wide range of structured systems of combat and physical discipline, traditionally practiced for self-defense, military training, spiritual development, and health enhancement. Over time, martial arts have evolved into multifaceted disciplines encompassing cultural heritage, philosophy, aesthetics, and competitive sports. Styles such as Wushu, Taekwondo, Karate, and Pencak Silat often blend rigorous physical training with meditative practices and ethical values, such as respect, discipline, and self-control (Ciaccioni et al., 2024; Stamenković et al., 2022). For instance, the Chinese martial art Wushu integrates spiritual and philosophical frameworks like Taoism and the yin-yang principle, which emphasize balance, fluidity, and harmony in motion (Han et al., 2021).

Wushu, also known broadly as Kung Fu, serves as a cultural embodiment of both traditional and modern Chinese martial arts. The word "Wushu" literally means "martial technique" in Chinese, and its modern practice includes both aesthetic routines and combat applications. As a sport, it emphasizes internal and external training, incorporating dynamic acrobatic movements and stylized attacks and defenses (Yangyang, 2023).

Lastly, in the Indonesian context, Pencak Silat carries similar significance where it is not just as a martial art, but to instill ethical behavior and leadership values (Schiff, 2022). These cultural factors serve as one of the driving forces to delve deeper on martial arts' movement complexity and intentionality.

Martial arts are particularly well-suited for computational modeling due to their non-repetitive, explosive motion characteristics. Techniques typically involve rapid accelerations, full-body coordination, and highly controlled limb trajectories, including spins, jumps, and strikes. Unlike locomotor activities such as walking or running, martial arts motions are structured into discrete sequences (e.g., forms, or "kata") that can be further segmented and analyzed computationally. This complexity has led to a growing interest in leveraging motion capture and biomechanical analysis to examine performance accuracy, prevent injury, and develop interactive systems such as virtual martial arts training platforms (Wan Idris et al., 2019).

2.1.2 Motion Capture

Motion capture (MoCap) is a technology used to digitally record, analyze, and reconstruct human or animal motion by tracking the position of body markers or sensors over time. In high-precision marker-based optical systems, reflective markers are attached to key anatomical landmarks and tracked by multiple infrared cameras. These cameras triangulate the markers' positions in 3D space, capturing fine-grained spatial-temporal data at high frame rates. Alternatively, markerless motion capture (MMC) uses computer vision techniques to analyze body movement without physical markers, allowing for more natural and comfortable motion tracking in various environments. The output will beta precise digital data that represents natural movement. The resulting output is a precise digital representation of movement, often structured as a time-series dataset of XYZ coordinates that reflects full-body kinematics with high resolution. MoCap is widely applied across disciplines including biomechanics, animation, sports science, and robotics, due to its ability to provide accurate, reproducible motion data.

As industries increasingly adopt digital tools under the umbrella of the Fourth Industrial Revolution, the application of motion capture has expanded rapidly, from gaming and industrial robotics to healthcare and martial arts education (Menolotto et al., 2020). The market for 3D MoCap systems alone was valued at over USD 222 million in 2023, reflecting its growing relevance in diverse sectors (SNS Insider, 2023). Specifically, in biomechanical research, MoCap enables the precise analysis of joint angles, limb trajectories, angular velocity, and postural shifts, essential for understanding complex physical movements like those found in Wushu, Kendo, and other martial arts disciplines.

In performance analysis and movement science, MoCap is valued for its non-invasive and objective capture of real-time joint behavior and coordination. Especially for martial arts, where rapid strikes, limb extensions, and rotational techniques demand accuracy and power, MoCap enables segmentation and visualization of movement phases that are difficult to track with the naked eye. MoCap data also serves as input for modeling skeletal animations, computing kinematic derivatives, and identifying motion transitions, which are tools that are vital for both scientific research and the development of real-time interactive systems such as virtual coaching platforms.

2.1.3 Motion Segmentation

Motion segmentation refers to the process of dividing continuous movement into smaller segments, where each segment represents a distinct phase of action. In martial arts, this is especially important because techniques such as kicks, strikes, and transitions often occur in rapid, discrete bursts that cannot be captured effectively as a single continuous action.

Segmenting these motions enables more focused analysis of specific techniques, facilitates feedback for performance improvement, and supports downstream applications such as classification, simulation, or automated coaching. Unlike cyclic motions like walking or running, martial arts actions are often non-repetitive, asymmetrical, and high in variation, which makes traditional segmentation approaches less effective.

This study adopts a feature-based segmentation framework, where physical cues from the motion data are used to detect transition points. This approach assumes that specific physical changes correspond to changes in motion phases. Three features are used, which are angular velocity, directional change, and volume change. These methods are later combined and evaluated together.

Angular Velocity

Angular velocity measures the rate of joint rotation, where it captures how fast a joint rotates over time. This is useful in martial arts, where quick kicks or shifts in the body often involve strong bursts of joint rotation. Compared to linear velocity, angular velocity is better suited for tracking rotations in joints such as the hips, knees, and ankles.

To compute angular velocity, joint angles between connected body segments are first calculated using the arccosine of the dot product between their corresponding vectors, as shown in Equation (1):

$$\boldsymbol{\theta_i^k = \cos^{-1}\left(\frac{u \cdot v}{\|u\|\|v\|}\right)} \tag{1}$$

Where:

- *i* represents the frame index
- *k* represents the angle index (1 through 6)
- *u, v* are the vectors between connected markers

Once the joint angles are known, angular velocity is computed using the backward difference method, with formula shown in Equation (2):

$$\dot{\boldsymbol{\theta}}_{\boldsymbol{i}} = \frac{\boldsymbol{\theta_i} - \boldsymbol{\theta_{i-1}}}{\mathbf{0.01}} \quad (2)$$

Here, 0.01 seconds is the time step, based on a frame rate of 100 frames per second.

For each frame, the angular velocities of all six angles are squared and summed to produce a single value z, shown in Equation (3):

$$\boldsymbol{z} = (\dot{\boldsymbol{\theta}}_{\boldsymbol{i}}^{\mathbf{1}})^{\mathbf{2}} + (\dot{\boldsymbol{\theta}}_{\boldsymbol{i}}^{\mathbf{2}})^{\mathbf{2}} + (\dot{\boldsymbol{\theta}}_{\boldsymbol{i}}^{\mathbf{3}})^{\mathbf{2}} + (\dot{\boldsymbol{\theta}}_{\boldsymbol{i}}^{\mathbf{4}})^{\mathbf{2}} + (\dot{\boldsymbol{\theta}}_{\boldsymbol{i}}^{\mathbf{5}})^{\mathbf{2}} + (\dot{\boldsymbol{\theta}}_{\boldsymbol{i}}^{\mathbf{6}})^{\mathbf{2}} \quad (3)$$

This value reflects the total rotational activity of the lower body at each frame.

Segmentation candidates are found by locating local minima in the z signal that occur after local maxima. A frame is only accepted as a segmentation point if the local minimum is below a dynamic threshold τ, with the formula shown in Equation (4):

$$\boldsymbol{\tau} = \boldsymbol{max}\,(\mathbf{0.1} \cdot \big(\boldsymbol{max}(\boldsymbol{z}) - \boldsymbol{min}\,(\boldsymbol{z})\big), \mathbf{100}) \quad (4)$$

This threshold ensures that only meaningful drops in rotational activity are considered, filtering out noise and shallow dips.

A frame is marked as a segmentation point if:

- z_i is a local maximum.
- z_j is a local minimum.
- Only accept z_j if z_j smaller than τ.

Direction Change

Direction change is based on the digital curvature of a marker's movement trajectory. In this study, we use the root marker focused on Left Middle Waist (LMWT) projected on the horizontal plane (XY) to track changes in movement direction. This method works well for detecting transitions like stepping forward, turning, or pausing.

To detect turning points, the method calculates digital curvature based on the angle between two temporal vectors—one pointing backward (Vector a) and one forward in time (Vector b). These vectors span a fixed offset of 15 frames (0.15 seconds), giving a 0.3-second analysis window.

Let p_i be the 2D (X & Y) position of the root marker at frame i, and L be the frame offset. The forward and backward vectors are shown in Equation (5) and (6):

$$\vec{a}_i = p_i - p_{i-L} \tag{5}$$

$$\vec{b}_i = p_{i+L} - p_i \tag{6}$$

The curvature at frame i is calculated as the cosine of the angle between these vectors, as shown in Equation (7):

$$C_i = \cos\theta_i = \frac{\vec{a}_i \cdot \vec{b}_i}{\|\vec{a}_i\|\|\vec{b}_i\|} \tag{7}$$

Values near 1 indicate straight motion, near 0 indicate a 90° turn, and below 0 signal sharp or reversing motion.

To avoid noise, frames are excluded if either vector has insufficient displacement, as shown in Equation (8):

$$\|\vec{a}_i\| < T \text{ or } \|\vec{b}_i\| < T \tag{8}$$

Where T = 30 mm is the minimum motion threshold.

A frame is marked as a segmentation point if:

- C_i is a local minimum.
- C_i falls below the curvature threshold $\tau (\tau = 0)$

Volume Change

This method segments motion by measuring changes in body posture through the volume of a 3D convex hull built from all marker positions at each frame. The convex hull volume expands during wide stances and contracts during compact ones, providing a global measure of spatial posture.

Let C_i be the convex hull volume at frame i, computed from valid 3D marker coordinates using MATLAB's *convhull()* function. The frame-to-frame volume difference is defined as:

$$D_i = |C_i - C_{i-1}| \tag{9}$$

As shown in Equation (9), D reflects posture transitions. Local minima in D often corresponds to short pauses or transition phases between movements.

A frame i is marked as a segmentation point if:

- D_i is a local minimum.

- Only accept D_i if D_i smaller than τ, and D_{i+1} bigger τ.

$$\tau = \boldsymbol{max}\left(\mathbf{0.23} \cdot \left(\boldsymbol{max}(\boldsymbol{D}) - \boldsymbol{min}\left(\boldsymbol{D}\right)\right), \mathbf{2} \times \mathbf{10^6}\right) \quad (10)$$

This adaptive threshold shown in Equation (10) ensures segmentation is responsive to motion scale while filtering out noise or shallow fluctuations.

Motion segmentation continues to play an essential role across various domains, from healthcare and sports to interactive systems and digital media. As motion capture becomes more widespread and detailed, segmentation methods that balance accuracy, efficiency, and interpretability remain crucial, especially in specialized fields like martial arts where movements are complex and labeled data is often limited.

2.1.4 Ground Truth as an Evaluation Metric

In the context of movement segmentation, ground truth refers to the set of frame indices that have been manually labeled or verified by experts as true transition points between motion phases. These labels are treated as the standard reference against which the performance of an algorithm is measured. Ground truth plays a critical role in validating the effectiveness of automated segmentation methods. Without it, there is no reliable benchmark to determine whether a predicted point correctly identifies a meaningful change in motion.

Ground truth data is typically derived from visual inspection, expert annotation, or consensus among multiple observers. In this study, the ground truth points were obtained through manual labeling of key motion transitions in martial arts sequences. After filtering for redundancy and enforcing a minimum temporal separation, this final ground truth set serves as the reference for performance evaluation.

To assess how well the predicted segmentation points align with the ground truth, we use a metric based on cosine similarity. This approach treats evaluation as a geometric matching problem between two binary vectors: one representing predictions, the other representing ground truth labels.

Let:

- A be a binary vector representing predicted segmentation points
- B be a binary vector representing the ground truth points
- Each element of A is 1 if a prediction matched a ground truth point (within ±100 frames), else 0

- ‖A‖ and ‖B‖ represent the Euclidean norms (i.e., square roots of the number of matched predictions and total ground truth points).

The cosine similarity is computed as follows:

$$\boldsymbol{Similarity} = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{\|\boldsymbol{A}\| \cdot \|\boldsymbol{B}\|} \tag{11}$$

Where:

- $A \cdot B$ is the total number of matched predictions (true positives)
- $\|A\|$ = √number of matched predictions
- $\|B\|$ = √number of ground truth points

This metric gives a single value between 0 and 1, where 1 indicates perfect alignment between predicted and ground truth points. Unlike precision or recall, cosine similarity treats segmentation performance as a measure of angular alignment in a high-dimensional binary space. It is especially useful in scenarios where over-segmentation or under-segmentation can skew traditional metrics.

By using this approach, the evaluation emphasizes overall alignment quality rather than penalizing minor deviations in timing. This is well-suited for motion capture tasks, where transition points may not always be detected at the exact frame but still correspond to semantically meaningful boundaries.

2.2 Literature Review

This section provides a comparative overview of existing research in motion segmentation, highlighting various methodologies and their applications. By examining these diverse approaches, this review aims to contextualize the present study's chosen methodology within the broader landscape of motion analysis, identify key challenges in segmenting complex human movements like martial arts, and demonstrate how prior work informs the adaptation and evaluation of the proposed framework.

Here we will compare relevant works of motion segmentation. We have 6 relevant papers, representing different approaches to motion segmentation and analysis, each offering unique strengths and insights that are pertinent to the goals of this thesis. These works range from foundational rule-based methods to advanced learning-based techniques and feature-driven approaches, providing a comprehensive backdrop against which the current research can be

positioned and evaluated. Table 2.1 shows the comparison of relevant papers segmented into its corresponding sections.

Table 2.1 Literature review from relevant papers

No.	Title	Author	Method	Object	Results	Limitations
1	Segmentation of Dancing Movement by Extracting Features from Motion Capture Data	Masafumi Sonoda et al. (2008)	Physical level segmentation of whole body dance motion. Uses three types of information: change of speed of body motion, change of direction of whole body motion, and change of size of the space produced by the body. Speed-based method traces the fastest marker. Direction-based method observes	Motion capture data for whole body motion during dance, specifically classical ballet and contemporary street dance (break dance).	Experiments show promising results for ballet and break dance. Integration of results from multiple methods improves recall (e.g., 0.59 to 0.85 for ballet, 0.11-0.44 to 0.64-0.69 for break dance). Method detected frames where dance steps changed.	Simple speed-based methods might detect irrelevant candidates. The method does not account for hierarchy of motion patterns; thus, it is not directly applicable to dances with hierarchical structures. Will not work well for detecting cyclic motions in repetitive actions. Precision for break dance was not always satisfactory. Does not use semantic-level information (meaning, style) of the dance.

			Root marker trajectory and uses digital curvature. Space-based method uses volume of 3D convex hull enclosing markers. Integrates candidates using logical OR and merging close/similar points.			
2	Fine-Grained Semantic Segmentation of Motion Capture Data using Dilated Temporal Fully-Convolutional Networks	N. Cheema et al. (2019)	Semi-automatic framework based on supervised machine learning techniques. Transforms motion capture sequences into a "motion image" and applies a dilated	Fine-grained semantic segmentation of human motion capture data. Examples include "standing," "begin left step," "left step in locomotion," "reach," and "turn (in standing)". Evaluated on	Outperforms two state-of-the-art models for action segmentation (ED-TCN, DilatedTCN) and a popular sequence modeling network (Bi-LSTM). Achieves 90.81% test accuracy	Existing (semi-)automatic solutions often require hand-crafted features or lack semantic annotations. Human-labeled annotation data suffers from inconsistencies. Kinematic-based methods are not versatile for diverse actions. Unsupervised methods lack

			temporal fully-convolutional neural network (DT-FCN) for image segmentation. Uses dilated temporal convolutions for large receptive fields. Model is fully-convolutional and handles variable length input sequences.	standing, walking, picking, placing, and turning actions.	without noise, and remains very robust under noisy and inaccurate training labels (e.g., almost 90% accuracy with 80% noisy labels). Training is magnitudes faster than RNN-based methods (~1 minute vs. ~40 minutes). Masking boundary frames improves performance (e.g., 92.42% accuracy with 11 frames masked vs. 90.81% without). Can distinguish	control and semantic labeling. Recurrent Neural Networks (RNNs) have exploding/vanishing gradient problems and parallelization difficulties.

					similar motion primitives like "start left step" and "left step".	
3	Skeleton-Based Action Segmentation with Multi-Stage Spatial-Temporal Graph Convolutional Neural Networks	Benjamin Filtjens, Bart Vanrumste, Peter Slaets (2022)	Proposes Multi-Stage Spatial-Temporal Graph Convolutional Neural Networks (MS-GCN). Replaces initial temporal convolutions with spatial graph convolutions and dilated temporal convolutions. Combines dilated temporal convolutions, spatial graph convolutions, and multiple refinement	Skeleton-based action segmentation for human movement analysis. Uses marker-based, marker-less optical, and inertial-based MoCap representations. Evaluated on diverse datasets including PKU-MMDv2, HuGaDB, LARa, FOG-GAIT (freezing of gait), and TUG (Timed Up-and-Go).	Statistically outperformed four strong baselines (Bi-LSTM, TCN, ST-GCN, MS-TCN) on four out of five datasets. Refinement stages significantly reduce over-segmentation errors (improves F1@50 scores). Explicitly modeling spatial hierarchy (graph convolutions) is beneficial, impacting sample-wise accuracy.	Action segmentation is more challenging than recognition due to simultaneous recognition and localization. Sliding-window methods have trade-offs between context and sensitivity to short actions. Predictions tend to over-segment at high temporal frequencies. Generic action segmentation approaches do not consider spatial hierarchy.

			stages. Uses acausal convolutions (considers past and future).		Dilated convolutions improve accuracy, and acausal convolutions perform better by utilizing future context.	
4	Unsupervised motion capture data segmentation based on topic model	Xiaoyan Hu et al. (2021)	Unsupervised motion segmentation method based on Sparse Topical Coding (STC) topic model. Builds a "motion vocabulary" from poses using hierarchical agglomerative clustering. Views motions as "motion documents" generated by a sliding	Motion capture data for unsupervised motion segmentation. Evaluated on CMU Mocap dataset (subject 86, other subjects) and HDM05 dataset (unknown subjects and motion types). Includes motions like walking, jumping, punching, leg kicks, jogging,	Achieves very high accuracy (average 94.53% on subject 86 CMU Mocap). Shows strong generalization ability across different subjects and datasets. Outperforms ACA and HACA in some cases. STC produces sparse and highly discriminabl	LDA (another topic model) may perform poorly on short texts/actions and lacks control over posterior sparsity. May not perform well on very short motions. Accuracy can vary between subjects. Quality of motion vocabulary is crucial.

			window. Extracts sparse topical codes using STC. Conducts spectral clustering on document codes to find segmentation points. Uses silhouette coefficient to determine the number of motion types (K).	bending knees.	e document codes. Can identify the same motion type across datasets and segment sub-actions (e.g., right/left leg kicks).	
5	Sensor Combination Selection Strategy for Kayak Cycle Phase Segmentation Based on Body Sensor Networks	Sen Qiu et al. (2022)	Automatic phase segmentation using inertial sensors (IMUs) and data fusion algorithms to calculate joint angles. Extracts 23 characteristic variables	Fine time phase segmentation of kayak rowing cycle. Divides the cycle into 8 phases (left/right blade entry, pull, exit, recovery). Investigates kayaker's	Precision increases with more motion information (more sensor data). Combination 8 (LUA-T + RUA-T, using 3 nodes) with SVM is the optimal	Traditional video methods have blind spots, are light-affected, less accurate, and inconvenient; manual sensor combination selection is complex for more nodes. Misclassification occurs at phase transitions due to

			from time and frequency domains in 20-sample windows. Evaluates nine different angle sequence combinations as input features and four machine learning algorithms: Decision Tree, SVM, KNN, Bagging Ensemble Learning. Applies PCA for dimension reduction. Uses 5-fold cross-validation.	technique and upper limb coordination. Conducted outdoor experiments with kayakers.	solution (98.1% accuracy, 96.6% Macro-PPV). SVM shows the highest accuracy among tested algorithms. Shoulder joint angle provides more information than elbow angle. IMU-based method is reliable (comparable to OptiTrack) and superior to optical systems for on-water sports. More convenient than manual video labeling. Useful for analyzing	training data strategy. Single-angle sequences are less effective. One-sided angle sequences lead to unstable performance due to handedness. Too many nodes reduce athlete comfort.

					paddling rhythm and technical characteristics.	
6	Martial Arts Moves Recognition Method Based on Visual Image	Husheng Zhou (2022)	Segments and extracts Wushu movement signals from visual images. Involves denoising using wavelet transform (e.g., 3-layer). Uses a sliding window data segmentation technique. Calculates net acceleration. Builds a Hidden Markov Model (HMM) for each martial arts movement. Baum-	Recognition and segmentation of Wushu (martial arts) movements from visual images. Focuses on improving accuracy and efficiency. Training data includes 8 martial arts movements: flat horse, lunge, punch, split, palm push, palm split, kick, spring.	Wavelet transform effectively denoises data (3-layer better than 2-layer; 17.5 dB higher SNR, 4.48 dB lower MSE). HMM-based model shows higher recognition accuracy than SVM, CNN, and BP neural network. For 1,000 samples, HMM achieves 99.75% accuracy (up to 1.50%p higher). For 4,000 samples, HMM	SVM easily overfits and locally optimizes, leading to inadequate accuracy. Baum-Welch single training can lead to local optimization and low recognition rates if only one dataset is used. Absence of prepared sample data raises concerns about flawed experimental results.

			Welch algorithm trains HMM parameters, enhanced with frequency weighted training and mean training methods. Viterbi algorithm identifies movements by determining optimal state sequence probability.		achieves 99.60% accuracy (up to 1.14%p higher). Frequency weighted training of Baum-Welch converges better. System is trustworthy and effective.	

Sonoda et al. (2008) introduced a feature-driven method combining three interpretable physical features: linear velocity, direction change, and convex hull volume. Their approach, although originally designed for dance motions, provides a compelling framework for martial arts due to its ability to detect transitions in expressive, non-cyclic movements without requiring training data. Unlike many learning-based methods, this approach is domain-agnostic, lightweight, and adaptable—offering a transparent mechanism to identify phase boundaries, which is critical when expert-labeled datasets are scarce or unavailable.

In contrast, Cheema et al. (2019) and Filtjens et al. (2022) applied supervised deep learning methods, including fully convolutional networks and graph-based models. While these methods demonstrate high segmentation accuracy (up to 90%+) and effective modeling of long-range dependencies, they are heavily dependent on large, labeled datasets, and require significant computational resources. Moreover, their black-box nature makes interpretation of

segmentation boundaries difficult, limiting their suitability in martial arts contexts where explainability and domain validation are key.

Unsupervised methods, such as Hu et al.'s (2021) STC-based topic model, show strong generalization and segmentation performance without labels, but they rely on clustering over motion vocabularies that may not perform well on very short or explosive motion types—common in martial arts. Similarly, sensor-based approaches like Qiu et al. (2022) (IMU-based phase recognition for kayaking) are accurate but unsuitable for martial arts applications due to wearability concerns, complexity of real-world deployment, and limited motion diversity in training.

Zhou's (2022) HMM-based method for Wushu recognition from visual signals highlights the viability of statistical modeling in martial arts, but its reliance on hand-crafted features and image-based preprocessing introduces limitations in 3D kinematic accuracy and sensor dependency.

Taken together, Sonoda et al.'s method offers the best balance between interpretability, flexibility, and domain suitability for martial arts motion, which features rapid, irregular, and coordinated full-body transitions. Its reliance on low-level physical descriptors—linear velocity, trajectory curvature, and spatial expansion—makes it highly adaptable to the non-periodic and explosive nature of martial arts. Additionally, the method's modularity allows for individual tuning of each feature, which aligns well with the biomechanical richness and segmentation challenges present in techniques like Wushu kicks. Therefore, this study adapts Sonoda's framework, extending its application from dance to martial arts, while integrating domain-specific refinements to suit the unique demands of the task.

CHAPTER III
RESEARCH METHODOLOGY

This chapter outlines the research methodology employed in this study. The motion capture data for this research will be obtained from the publicly accessible online archive of the National University of Singapore, which hosts a diverse collection of MoCap recordings, including dance, sports, and martial arts performances. From this extensive dataset, our study will focus specifically on Wushu kicking motions. The raw 3D coordinate data was preprocessed and visualized using MATLAB, which provides robust tools for data filtering, coordinate transformation, and 3D rendering. Utilizing MATLAB enhances the accuracy and interpretability of our motion analysis and differentiates our methodology from that of Sonoda et al., who employed alternative computational techniques.

Following the visualization process, we implemented three segmentation methods adapted from Sonoda et al.'s framework: segmentation based on angular velocity, changes in motion direction, and changes in volume via convex hull. After key frames are segmented using these features, the reconstructed 3D skeleton visualizations will be used to illustrate the resulting motion phases based on the segmentation. The segmentation results will then be compared against ground truth data, which are collected from experienced martial arts respondents. The performance of the segmentation methods will be quantitatively evaluated and compared with the ground truth using cosine similarity to measure accuracy.

Figure 3.1 shows the research methodology flowchart, where the methodology are broken down into 3 sections: **Preprocessing**, **Segmentation**, and **Evaluation**.

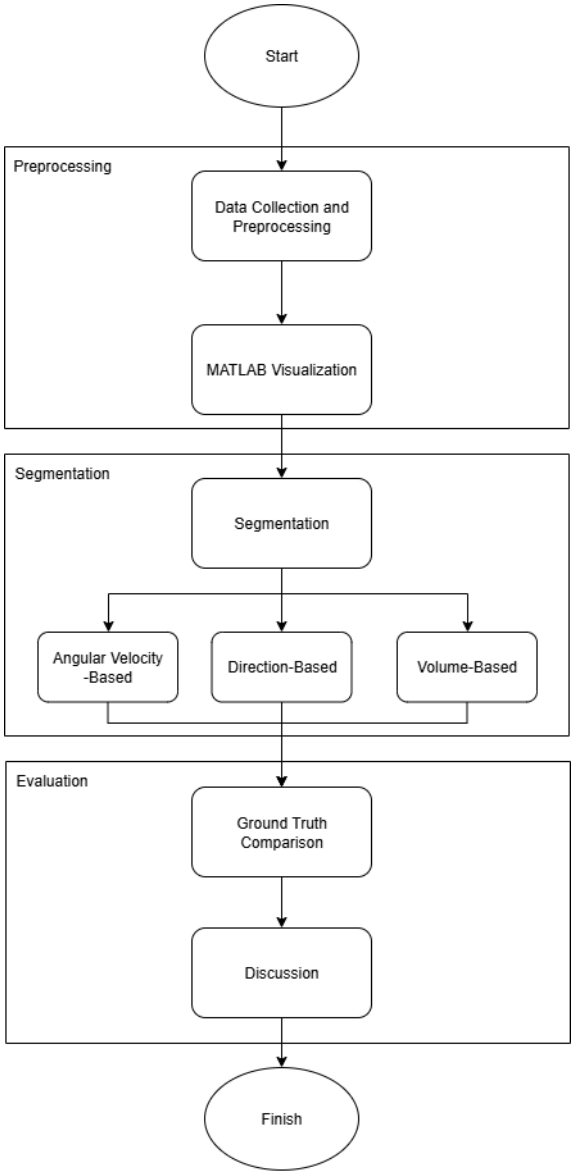


Figure 3.1 Research methodology flowchart

3.1 Preprocessing

3.1.1 Collecting and Preprocessing Motion Capture Data

To ensure the motion capture data is properly structured for analysis, it was preprocessed before being imported into MATLAB. The collected Wushu motion capture data consists of X, Y, and Z coordinates for multiple body markers, recorded as time-series data. Since the raw data is initially stored in .txt format, it will be converted into an Excel (.xlsx) file for better organization and accessibility. This conversion allows easier data manipulation and integration with preprocessing scripts. Additionally, the Excel file will be modified to combine the X, Y, and Z coordinates for each marker into a single cell, simplifying the data structure for efficient parsing in MATLAB.

Converting Coordinate Data into Excel

The Wushu coordinate data, consisting of X, Y, and Z values for each body marker, will be initially collected in *.txt* format and subsequently converted to *.xlsx* to facilitate easier management and analysis.

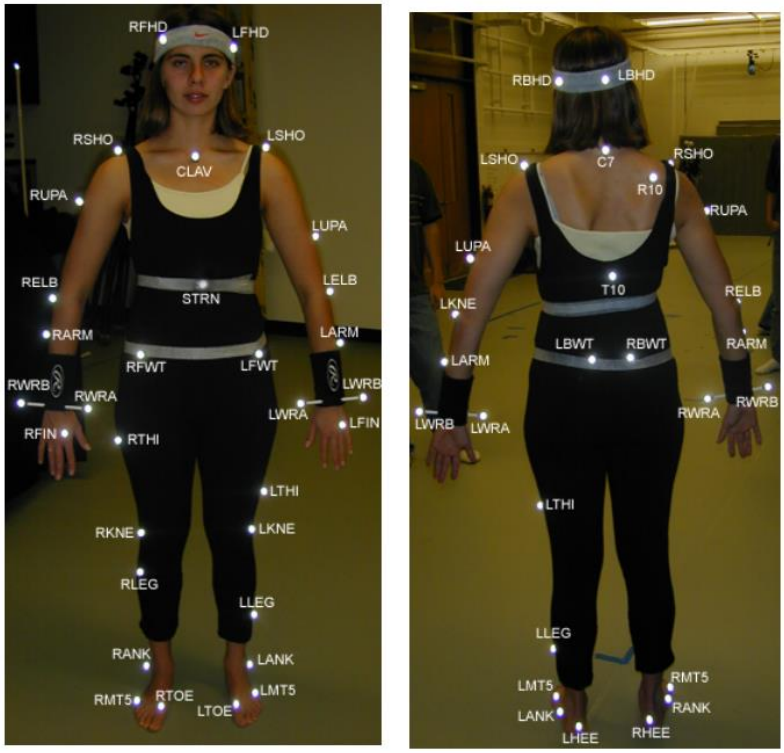


Figure 3.2 Full-body marker position and labels

Source: (Carnegie Mellon University, n.d.)

Image in Figure 3.2 shows a brief visualization of marker placement in the human body as well as its labels. It is important to note that the marker placement guide shown in Figure 3.2 does not fully reflect the configuration used in our martial arts dataset, due to some differences in marker placement. For instance, our dataset includes four markers to represent the shoulders, which are LFSH, LBSH, RFSH, and RBSH. Figure 3.2 on the other hand, only uses two markers, LSHO and RSHO, to represent the same regions. Table 3.1 shows the raw coordinate data, recently converted from *.txt* to *.xlsx.*

Table 3.1 Wushu kicking motion coordinates for marker ARIEL from frame 1 to 9

Frames	ARIEL X	ARIEL Y	ARIEL Z
1	209.772	1649.082	-183.245
2	208.051	1648.934	-183.377

3	206.286	1648.746	-183.427
4	204.466	1648.519	-183.403
5	202.625	1648.271	-183.335
6	200.72	1648.036	-183.244
7	198.767	1647.824	-183.173
8	196.756	1647.652	-183.122
9	194.689	1647.502	-183.076

Implementing Visual Basic for Applications (VBA) into Excel

To facilitate the seamless import of coordinate data into MATLAB, modifications to the table structure are necessary. Specifically, the X, Y, and Z coordinates for each marker must be combined into a single cell to streamline data processing. To achieve this, we implement Visual Basic for Applications (VBA) within the Excel file, automating the restructuring process. The VBA script inserts new empty columns after the first column and subsequently repeats this pattern after every three columns. These newly added columns serve as dedicated spaces for storing the combined X, Y, and Z coordinate values for each marker, ensuring a more structured and efficient data format for MATLAB processing. Figure 3.3 VBA code for adding new columns automatically shows the VBA code used to add the new columns, and Table 3.2 Wushu kicking motion coordinates after VBA codeis the Excel result after implementing the VBA code.

```
Sub AddColumns()
    Dim ws As Worksheet
    Dim col As Long
    Dim lastCol As Long

    Set ws = ThisWorkbook.Sheets("Sheet1")

    lastCol = ws.Cells(1, ws.Columns.Count).End(xlToLeft).Column

    For col = lastCol To 1 Step -1
        If (col - 1) Mod 3 = 0 Then
            ws.Columns(col + 1).Insert
        End If
    Next col
End Sub
```

Figure 3.3 VBA code for adding new columns automatically

Table 3.2 Wushu kicking motion coordinates after VBA code

Frames		ARIEL X	ARIEL Y	ARIEL Z
1		209.772	1649.082	-183.245
2		208.051	1648.934	-183.377
3		206.286	1648.746	-183.427
4		204.466	1648.519	-183.403
5		202.625	1648.271	-183.335
6		200.72	1648.036	-183.244
7		198.767	1647.824	-183.173
8		196.756	1647.652	-183.122
9		194.689	1647.502	-183.076

Concatenating Values into New Cells

Once the new columns have been created, the next step is to populate them with the combined coordinate data. This is accomplished using an Excel formula that employs the ampersand (&) operator to concatenate the X, Y, and Z coordinate values into a single cell. The formula is structured to enclose the combined coordinates within parentheses ((...)), ensuring clarity and consistency in formatting. Each X, Y, and Z value is separated by a comma (","), making the data easily parse-able in MATLAB for further analysis. Figure 3.4 shows the Excel formula used for concatenating each marker's coordinate data. To efficiently apply the formula across the dataset, the filled cells can be dragged down to extend the calculations across all frames. Once the data is structured correctly, the filled columns can be copied and pasted into the next empty column, ensuring all markers are processed in a uniform manner. Finally, each new marker column is manually renamed to maintain organizational clarity and consistency in the dataset. The result of concatenation can be seen in Table 3.3.

```
=”(“ & C2 & “,” & D2 & “,” & E2 & “)”
```

Figure 3.4 Excel formula used for concatenation

Table 3.3 Wushu kicking motion coordinates after concatenation

Frames	ARIEL	ARIEL X	ARIEL Y	ARIEL Z
1	(209.772, 1649.082, -183.245)	209.772	1649.082	-183.245
2	(208.051, 1648.934, -183.377)	208.051	1648.934	-183.377
3	(206.286, 1648.746, -183.427)	206.286	1648.746	-183.427
4	(204.466, 1648.519, -183.403)	204.466	1648.519	-183.403
5	(202.625, 1648.271, -183.335)	202.625	1648.271	-183.335
6	(200.72, 1648.036, -183.244)	200.72	1648.036	-183.244
7	(198.767, 1647.824, -183.173)	198.767	1647.824	-183.173
8	(196.756, 1647.652, -183.122)	196.756	1647.652	-183.122
9	(194.689, 1647.502, -183.076)	194.689	1647.502	-183.076

3.1.2 Processing Data and Visualizing in MATLAB

Once the modified Excel file is structured correctly, it is imported into MATLAB for pre-processing, visualization, and analysis. This step is crucial as it ensures that the motion capture data is formatted appropriately for segmentation and further feature extraction.

Loading and Structuring the Data

The first step in MATLAB is to load the motion capture data using the readcell function. This approach retains the structure of the dataset, including column headers, which represent body parts. The script then identifies the columns containing combined (X, Y, Z) coordinates, which are formatted in Excel during pre-processing. This step is essential for mapping motion markers correctly to their respective anatomical positions, which is critical for segmentation and visualization.

Extracting and Converting Motion Data

Once the body parts are identified, the script extracts coordinate values for each frame. The data is stored as separate matrices for X, Y, and Z coordinates, making it easier to manipulate and analyze later.

Parsing Coordinates

A loop iterates through each frame of the motion capture data, extracting the combined coordinate values from each row. The strsplit function is used to separate the X, Y, and Z components, which are subsequently converted to numeric values. However, the original coordinate order does not conform to standard axis conventions, leading to a misaligned or incorrectly oriented posture when visualized. To ensure accurate skeleton visualization, the coordinate axes are rearranged into a custom order. In our case, the axes are reordered from the original X-Y-Z format to Y-Z-X, which aligns the skeleton upright and maintains correct anatomical orientation during rendering. Figure 3.5 shows a MATLAB code snippet in pseudocode where we implement axes rearrangement.

```
FOR each frame_row from 2 to num_frames + 1:
    FOR each body part index i from 1 to num_parts:
        col_index ← combined_cols[i]
        raw_data ← orang[frame_row, col_index]
        data_string ← convert cell to string (raw_data)

        // Remove parentheses
        data_string ← remove '(' from data_string
        data_string ← remove ')' from data_string

        // Split string by comma into coordinate components
        coords ← split data_string by ','

        IF coords has exactly 3 elements:
            y_coords_all[frame_row - 1, i] ← convert coords[0] to number
            z_coords_all[frame_row - 1, i] ← convert coords[1] to number
            x_coords_all[frame_row - 1, i] ← convert coords[2] to number
        END IF
    END FOR
END FOR
```

Figure 3.5 MATLAB code snippet to parse coordinates

Interpolating Missing Values

Motion capture data—such as the Wushu kicking motion used here—contains missing or zero values, which can disrupt accurate segmentation and visualization. To mitigate this, spline interpolation is applied to estimate and fill in the missing values, which can ensure continuity and preserving the smoothness of the motion trajectory.

Defining Skeletal Structure for Visualization

To construct a meaningful 3D skeletal representation, the study defines a set of anatomical connections between motion capture markers. These manually specified connections represent joint relationships that approximate the human skeleton, allowing for a coherent and interpretable depiction of body movement.

Visualizing the 3D Skeleton

A 3D plotting environment is initialized to prepare for skeleton visualization. This includes axis configuration, marker and line creation, and an optional surface for enclosing the body.

Animating the 3D Skeleton

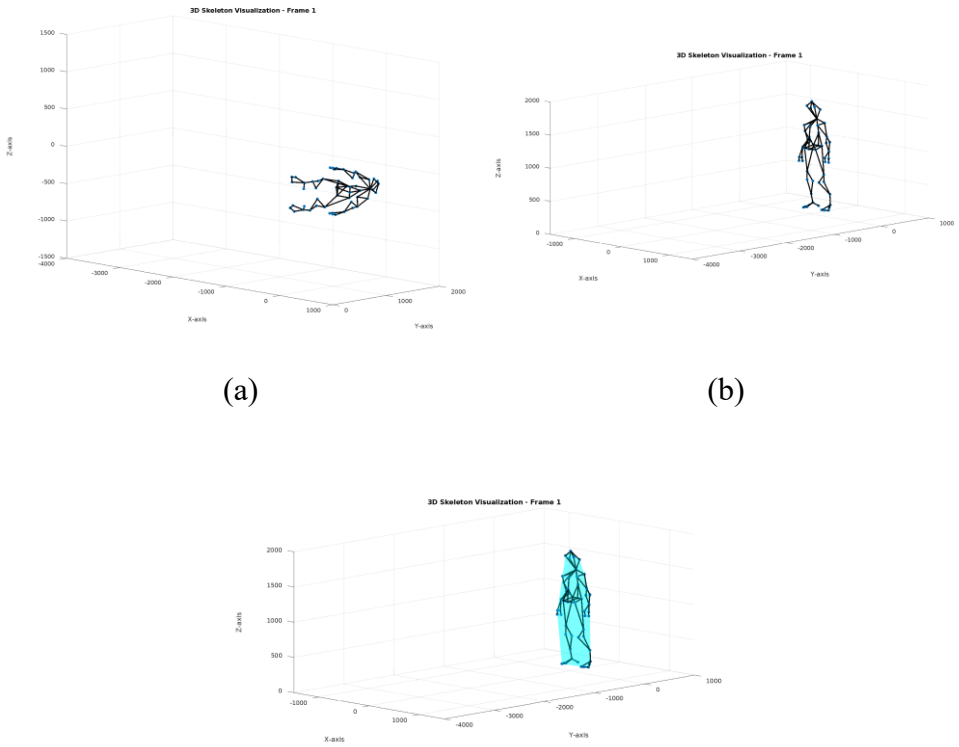
The animation loop visualizes the skeleton over time by updating marker positions for each frame of motion data. This dynamic update creates the effect of a moving 3D skeleton. The loop will pause every 0.01 seconds, which represents 100 frames per second.

Figure 3.6 shows three stages of skeleton visualization for Frame 1.

(a) shows the initial result without adjusting the coordinate axes. The skeleton appears to be lying down due to misaligned axes, making the posture hard to interpret.

(b) displays the corrected visualization after reordering the axes to Y-Z-X, making the skeleton upright and more accurate to its real pose.

(c) shows the same corrected frame with a convex hull applied. The convex hull outlines the body's volume and is used in the segmentation method based on volume change across frames.



(a) (b)

(c)

Figure 3.6 Skeleton visualization

3.2 Segmentation

Before identifying segmentation candidates, we established three guiding conditions to ensure consistency and relevance in the segmentation results:

a. Does each method's threshold closely resemble the principles outlined in Sonoda's original approach?
b. Is the total number of segmentation candidates for each method fewer than 22?
c. For each set of 11 kicks, does every kick have at least one corresponding segmentation candidate?

These conditions serve as practical guidelines during implementation to help achieve high-quality segmentation outcomes.

Condition (a) is intended to maintain conceptual fidelity to Sonoda et al.'s method, especially with respect to thresholding. Since Sonoda's original paper does not specify exact numerical thresholds for the angular velocity and volume difference methods, we retain the flexibility to define our own thresholds—while avoiding excessive deviation that would fundamentally alter the original methodology.

Condition (b) aims to prevent over-segmentation by capping the number of candidates per method. This ensures a manageable number of meaningful segmentation points, ideally allowing no more than two per segmented region.

Condition (c) ensures that each of the 11 kicking motions is represented by at least one segmentation point. This is achieved by carefully adjusting the threshold values to capture all key motion transitions.

These conditions are not strict requirements; the final segmentation results do not need to satisfy all of them perfectly. Rather, they act as reference points during method adaptation and tuning, allowing for modifications to Sonoda's framework while preserving its core structure and intent.

3.2.1 Segmentation by Angular Velocity

This research opts to employ angular velocity for motion segmentation instead of linear velocity, as originally proposed by Sonoda et al. Angular velocity is particularly well-suited for analyzing complex martial arts movements, where rapid and distinct rotational movements of body segments serve as key indicators of transitions between motion primitives. As demonstrated in prior studies, such as from Fod et al. (2002), angular velocity provides a solid feature for detecting dynamic changes and segment boundaries due to its heightened sensitivity to rotational motion—an aspect that linear position data alone may not sufficiently capture.

Prior to segmentation, several preprocessing steps are conducted to prepare the motion capture data. First, a subset of key markers is identified based on their relevance and high mobility. For Wushu, which emphasizes dynamic leg movements, the focus is placed on lower-limb markers. Based on these, six joint angles are computed for Wushu by forming vectors between body markers across all 2,378 motion frames. Each angle is calculated using the arccosine formula, which enables a quantitative assessment of the rotational relationships between connected body parts over time.

Once the joint angles are obtained for all frames, angular velocity is computed using the backward difference method: each angle is subtracted from its value in the previous frame and divided by the fixed time interval of 0.01 seconds (corresponding to 100 frames per second). For each frame, the squared angular velocities from all joint angles are summed to produce a single scalar value, denoted as z, which represents the overall rotational activity of the lower body.

To identify motion segment boundaries, local maxima in the smoothed signal are first detected to indicate bursts of joint rotation. From each of these maxima, the algorithm traces forward to find the next local minimum, representing a pause in motion. Only those local minima occurring after a maximum are considered segmentation candidates and are accepted only if they fall below an adaptive prominence threshold.

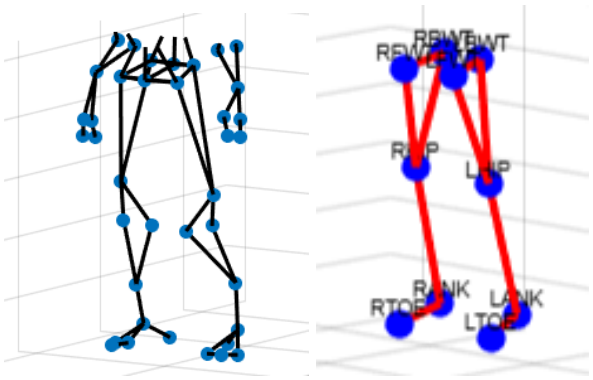
Selecting Key Markers

Given the complexity of full-body motion capture data, analyzing all available markers can be computationally intensive and may introduce unnecessary noise. To streamline the analysis, we focus on ten key markers located on both legs from the Wushu data:

- Right leg:
  - RFWT (Right Front Waist)
  - RBWT (Right Back Waist)
  - RHIP (Right Hip)
  - RANK (Right Ankle)
  - RTOE (Right Toe)
- Left leg:
  - LFWT (Left Front Waist)
  - LBWT (Left Back Waist)
  - LHIP (Left Hip)
  - LANK (Left Ankle)
  - LTOE (Left Toe)

These markers were selected due to their high mobility and prominent role in Wushu kicking motion. To preserve anatomical relevance, we manually define the connections between these markers to form a simplified yet representative model of the human leg. The

comparison between the main skeleton visualization (a) and the key selected markers (b) are shown in Figure 3.7.



(a) (b)

Figure 3.7 Leg structure comparison

Computing Joint Angles

Based on the marker connections, six angles are gathered from Wushu's key markers:

- $\theta_i^1 = \angle(RHIP - RBWT, RANK - RHIP)$
- $\theta_i^2 = \angle(RHIP - RFWT, RANK - RHIP)$
- $\theta_i^3 = \angle(RANK - RHIP, RTOE - LANK)$
- $\theta_i^4 = \angle(LHIP - LBWT, LANK - LHIP)$
- $\theta_i^5 = \angle(LHIP - LFWT, LANK - LHIP)$
- $\theta_i^6 = \angle(LANK - LHIP, LTOE - LANK)$

After determining the angles, we can then calculate and find the angle value using arccosine formula. The formula will be repeated for all six angles on one frame and repeated again for all 2,378 frames. Figure 3.8 shows a MATLAB code snippet for Equation (1), which was shown in 2.1.3 Motion Segmentation.

```
Function ANGLE_BETWEEN(u, v):
    // Computes the angle (in radians) between two 3D vectors u and v
    // using the arccosine of their normalized dot product

    dot_product ← dot(u, v)
    norm_u ← norm(u)
    norm_v ← norm(v)

    cosine_theta ← dot_product / (norm_u * norm_v)

    // Clamp value to avoid numerical errors outside valid domain of arccos
    cosine_theta ← max(min(cosine_theta, 1), -1)

    angle ← arccos(cosine_theta)

    Return angle
End Function
```

Figure 3.8 MATLAB code snippet to compute joint angles

Computing Angular Velocity

After having all angles from all frames, we can then calculate each angle's angular velocity. Since we have all the angles, this can be achieved using backward difference method—by implementing subtraction between the current frame and its previous frame and divided by the time interval of 0.01 seconds (100 frames per second). Figure 3.9 shows a MATLAB code snippet for Equation (2), which was shown in 2.1.3 Motion Segmentation.

```
Input:
- theta_all: matrix of joint angles (in radians) for each frame and joint
- dt: time interval between frames (0.01 seconds)

Output:
- theta_dot_all: matrix of angular velocities

Initialize theta_dot_all[num_frames, 6] ← NaN

FOR frame_idx ← 2 TO num_frames DO
    FOR angle_idx ← 1 TO 6 DO
        IF theta_all[frame_idx, angle_idx] AND theta_all[frame_idx – 1,
angle_idx] are valid THEN
            theta_dot_all[frame_idx, angle_idx] ←
                (theta_all[frame_idx, angle_idx] – theta_all[frame_idx – 1,
angle_idx]) / dt
        END IF
    END FOR
END FOR
```

Figure 3.9 MATLAB code snippet to compute angular velocity

Now that all of the frame's angular velocity has been discovered, we can then square and sum the four angles for one frame to be our frame's representative for their angular velocity, denoted as z. Figure 3.10 shows a MATLAB code snippet for Equation (3), which was shown in 2.1.3 Motion Segmentation.

```
Input:
- theta_dot_all: matrix of angular velocities (rows: frames, columns: joints)

Output:
- z: column vector containing sum of squared angular velocities per frame

Procedure:
FOR each frame i from 1 to num_frames DO
    z[i] ← sum of squares of all values in theta_dot_all[i, :]
END FOR
```

Figure 3.10 MATLAB code snippet to sum and square angular velocity

Identifying Segmentation Candidates

To identify motion segment boundaries, this method analyzes the scalar angular velocity signal z, which represents the sum of squared angular velocities from all joint angles. Local maxima in the z signal corresponds to bursts of rotational activity, often indicating the peak of a dynamic movement phase such as a Wushu kick.

Starting from each detected local maxima z_i, the algorithm traces forward through the signal to find the next local minima z_j, representing a potential pause or deceleration phase in the motion cycle. To ensure only significant pauses are considered, an adaptive threshold τ is applied for both local minima and maxima. The local minima are only accepted as a segmentation point if it satisfies $z_j < \tau$, which helps eliminate shallow or noisy minima.

The threshold τ is defined adaptively based on the range of the z signal. Figure 3.11 shows a MATLAB code snippet for Equation (4), which was shown in 2.1.3 Motion Segmentation.

```
Input:
- z_range: range of values in vector z (i.e., max(z) – min(z))

Output:
- tau: threshold value for segment detection

Procedure:
tau ← maximum of (0.1 × z_range) and 100
```

Figure 3.11 MATLAB code snippet for Equation (4)

Frames are selected as segmentation candidates if they satisfy:

- z_i is a local maximum: $z_{i-1} < z_i > z_{i+1}$
- z_j is a local minimum: $z_{j-1} > z_j < z_{j+1}, j > i$
- Only accept z_j if $z_j < \tau$, where τ is defined adaptively.

This formulation ensures that the threshold dynamically adjusts to the overall magnitude of movement in the motion sequence. The term $0.1 \cdot (\max(z) - \min(z))$ represents a percentage (10%) of the full range of the sum of squared angular velocity values across all frames, enabling the threshold to scale proportionally with the variation present in the sequence.

The use of the max function introduces a lower bound of 100, which prevents the threshold from falling below a meaningful minimum. This safeguard is particularly useful in sequences with low overall motion, as it reduces the likelihood of false segment detections caused by minor fluctuations or background noise in the angular velocity signal. Figure 3.12 shows a pseudocode for identifying angular velocity's segmentation points.

```
Input:
- z: vector of summed squared angular velocities per frame
Output:
- seg_points: array of frame indices selected as segmentation points
Procedure:

1. Compute range of z, ignoring NaN values:
   z_range ← max(z) – min(z), omitting NaNs

2. Define threshold:
   tau ← max(0.1 × z_range, 100)

3. Identify peaks in z:
   peak_locs ← indices of peaks in z with:
       - minimum prominence ≥ tau
       - minimum distance between peaks ≥ 10 frames

4. Identify local minima:
   min_locs ← indices of peaks in (–z), i.e., minima of z

5. Initialize empty list of segmentation points:
   seg_points ← []

6. For each peak in peak_locs:
   - Find the next minimum after the peak: next_min ← first min_locs >
peak_locs[i]
   - If next_min exists and z[next_min] < tau:
       - Append next_min to seg_points

End Procedure
```

Figure 3.12 MATLAB code snippet for angular velocity's segmentation

3.2.2 Segmentation by Change of Direction

This method segments martial arts motions by analyzing directional changes within the horizontal movement plane, defined by the side-to-side (X) and front–back (Y) axes. The Left Middle Waist (LMWT) marker is selected for both Wushu, as the root reference due to its central position near the body's center of mass. By projecting the root marker's trajectory onto the X–Y plane, the method calculates digital curvature to measure how sharply the body changes direction at each frame.

For every frame *i*, digital curvature is computed based on the angle between two temporal vectors—one pointing from a past frame to the current frame, and one from the current frame to a future frame. A frame is selected as a segmentation point if it is a local minimum in curvature and its value falls below a threshold (i.e., a strong turning motion). Additionally, only frames with sufficient displacement in both directions are analyzed, filtering out noise during stationary or low-motion phases. All steps are implemented in MATLAB.

Selecting Root Marker

To track directional changes, the Left Middle Waist (LMWT) marker is used as the root reference. Although the name implies a left-side placement, visual validation confirms LMWT lies near the frontal center of the waist, aligned with the Right Middle Waist (RMWT) marker at the back. This central position allows LMWT to effectively represent the body's horizontal trajectory and orientation.

Projecting Motion and Defining Vectors

For each frame *i*, two horizontal vectors are constructed:

```
Input:
- x_coords_all, y_coords_all: coordinate matrices for all frames
- root_idx: index of the root marker (e.g., LMWT)
- i: current frame index
- L: frame offset (e.g., 15)
Output:
- a: vector from past frame to current frame
- b: vector from current frame to future frame
Procedure:
a ← [
    x_coords_all[i] – x_coords_all[i – L],
    y_coords_all[i] – y_coords_all[i – L]
]

b ← [
    x_coords_all[i + L] – x_coords_all[i],
    y_coords_all[i + L] – y_coords_all[i]
]
```

Figure 3.13 MATLAB code snippet for Equation (5) and (6)

Figure 3.13 shows a pseudocode to define two vectors, where the formula can be seen in 2.1.3 Motion Segmentation. Here, p_i denotes the 2D (x, y) position of the LMWT marker at frame i, and L (15 frames) is a fixed offset. At a sampling rate of 100 frames per second, this offset corresponds to a 0.15-second interval in both forward and backward directions, effectively analyzing directional changes over a 0.3-second window. Vector a spans from a past frame to the current frame, while vector b spans from the current frame to a future frame.

Computing Digital Curvature

The cosine of the angle between vectors a and b defines the digital curvature C_i:

```
Input:
- a, b: 2D vectors representing motion before and after the current frame

Output:
- curvature[i]: cosine of the angle between vectors a and b

Procedure:
curvature[i] ← dot(a, b) / (‖a‖ × ‖b‖)
```

Figure 3.14 MATLAB code snippet for Equation (7)

Figure 3.14 shows a pseudocode for Equation (7) used to calculate digital curvature, with its formula shown in 2.1.3 Motion Segmentation.

This value ranges from −1 to 1, where:

- $C_i \approx 1$: straight motion
- $C_i \approx 0$: 90° turn
- $C_i \ll 0$: sharp or reversing motion

The curvature is computed only when both vectors meet the displacement threshold T.

Eliminating Low-Motion Frames

As proposed by Sonoda et al., to eliminate noise from near-stationary periods, any frame is excluded from curvature analysis if:

```
Input:
- a, b: 2D vectors representing motion before and after the current frame
- T: minimum displacement threshold

Condition:
IF ‖a‖ < T OR ‖b‖ < T THEN
    // Skip curvature computation for this frame
    Continue to next frame
END IF
```

Figure 3.15 MATLAB code snippet for Equation (8)

Figure 3.15 shows the minimum displacement threshold in pseudocode, which came from Equation (8) shown in 2.1.3 Motion Segmentation. T (30 mm) is a minimum displacement threshold where computation will not proceed if either vector is less than T or less than 30mm/3cm. This will ensure that only meaningful motion changes are segmented.

Identifying Segmentation Candidates

Unlike Sonoda et al., who detect segmentation points using local maxima, our method uses local minima due to a difference in curvature visualization. Sonoda's curvature plot starts near −1, producing upward spikes during turns. In contrast, our curvature starts at 1, with turns appearing as downward dips. To match this interpretation, we identify segmentation points where curvature drops below 0, corresponding to 90° or reversing motions in the horizontal plane. the pseudocode to identify segmented points for this method.

```
Input:
- curvature: array of digital curvature values per frame
- L: frame offset used in vector calculation
- curv_threshold: curvature cutoff (e.g., 0)

Output:
- seg_points_DirCha: list of selected segmentation frame indices

Procedure:

Initialize:
- seg_points_DirCha ← empty list
```

Figure 3.16 shows the pseudocode to identify segmented points for this method.

Figure 3.16 MATLAB code snippet for digital curvature's segmentation

Frames are selected as segmentation candidates if they satisfy:

- C_i is a local minimum: $C_{i-1} > C_i < C_{i+1}$
- $C_i < \tau$: falls below the curvature threshold $\tau\,(\tau = 0)$

3.2.3 Segmentation by Change of Volume

This segmentation approach utilizes the temporal variation in spatial volume produced by the body during motion, measured through the volume of a 3D convex hull formed from all marker positions in each frame. The convex hull volume serves as a global descriptor of body posture, expanding during wide or extended poses and contracting during compact stances. By analyzing the changes in this volume over time, the method detects transition points that signify meaningful posture shifts in martial arts movements.

Segmentation points are determined where the convex hull volume shows a significant change, crossing a predefined threshold. These points typically reflect moments of reduced body movement followed by expansion—characteristic of phase transitions in dynamic techniques. To ensure accuracy, redundant detections caused by minor fluctuations are filtered based on temporal distance and spatial similarity.

Constructing the Convex Hull

To analyze the spatial configuration of the body throughout motion, a 3D convex hull is constructed at each time frame using all available motion capture markers. The convex hull is defined as the minimal volume that encloses a given set of points in three-dimensional space. It effectively captures the body's spatial footprint and changes in posture over time.

Let C_i denote the convex hull volume at frame i. For each frame, the set of valid marker coordinates is extracted and filtered to remove any invalid (NaN) entries. The volume is then computed using MATLAB's *convhull()* function.

Calculating Volume Change

To detect transitions in movement, we calculate the frame-to-frame difference in convex hull volume. This difference, denoted D_i, captures the magnitude of spatial change from one frame to the next and is defined as:

This absolute value is computed across the sequence of convex volumes. In the implementation, the *diff()* function is used to obtain the differences, followed by an *abs()* to

compute magnitude. Equation (9) is used to calculate volume difference, represented in pseudocode in Figure 3.17:

```
Input:
- convex_volumes: array of convex hull volumes for each frame

Output:
- volume_diff: array of absolute volume changes per frame

Procedure:
1. Compute frame-to-frame difference:
   volume_diff ← absolute value of difference between successive convex_volumes

2. Extend to match original frame count:
   Append volume_diff[end] to the end of volume_diff
```

Figure 3.17 MATLAB code snippet for Equation (9)

Identifying Segmentation Candidates

To identify motion segment boundaries, this method analyzes the volume difference signal D, derived from the absolute frame-to-frame change in the 3D convex hull volume constructed from body markers. The convex hull volume reflects the spatial extent of the body posture at each frame, expanding during wide stances and contracting during compact transitions.

Local minima in the signal D often correspond to postural pauses, low-motion frames, or transitional moments, making them suitable candidates for segmentation. These dips typically occur between periods of larger volume changes, such as the transition between preparatory and execution phases of a dynamic martial arts technique.

Starting from each detected local minimum D_i, the method applies a two-part condition to filter only the most meaningful segmentation points. Specifically, the local minimum is accepted as a candidate if it satisfies:

Frames are selected as segmentation candidates if they satisfy:

- D_i is a local minimum: $D_{i-1} > D_i < D_{i+1}$
- Only accept D_i if $D_i < t$, and $D_{i+1} >$ t, where τ is defined adaptively.

This ensures that the selected point is a pronounced dip in the signal that is immediately followed by a burst of increased volume change, which indicates a phase transition.

The threshold t is computed adaptively, based on the overall range of the volume difference signal:

```
Input:
- volume_diff: array of absolute frame-to-frame convex hull volume changes
- num_frames: total number of motion frames

Output:
- seg_points_VolCha: list of frame indices selected as segmentation points

Procedure:

1. Compute adaptive threshold:
   t ← max(0.23 × range of volume_diff, 2 × 10^6), ignoring NaNs

2. Initialize:
   seg_points_VolCha ← empty list

3. For each frame i from 2 to num_frames – 1:
   IF volume_diff[i] is a local minimum AND
      volume_diff[i] < t AND
      volume_diff[i + 1] > t THEN

      Append i to seg_points_VolCha

   END IF
END FOR
```

Figure 3.18 MATLAB code snippet for volume diff segmentation and Equation (10)

Figure 3.18 shows the pseudocode for identifying segmentation candidates and Equation (10) that represents the adaptive threshold. This formulation ensures that the threshold dynamically adjusts to the overall magnitude of movement in the motion sequence. The term $0.23 \cdot (\max(D) - \min(D))$ represents a percentage (23%) of the full range of volume differences across all frames, providing adaptiveness to different movement intensities.

The use of the max function introduces a minimum bound of 2×10^6, preventing the threshold from becoming too small in low-motion sequences. This constraint helps reduce the risk of detecting shallow local minima caused by noise or minor fluctuations in posture.

3.2.4 Integrating Segmentation Method Results

In the approach proposed by Sonoda et al., temporally adjacent candidate frames are consolidated by retaining only the earliest frame within a close time window to avoid redundant segmentation points. In this study, a similar strategy is employed: segmentation candidates that fall within a 15-frame window are grouped, and the earliest frame index of each group is used as the final segmentation point.

Before integrating the results from the three segmentation methods, a postprocessing step is applied to each of the raw segmentation candidates. Specifically, if multiple frames occur

within a 50-frame window, they are grouped, and only the earliest frame in each group is retained as the representative segmentation point. The 50-frame window was selected to maintain a final count of segmentation points that closely approximates the expected number of motion transitions, which is 11 based on the number of Wushu kicks in the dataset. This ensures that each candidate represents a distinct motion transition and avoids redundant detections. To maintain transparency in performance evaluation, the raw results of each method are presented independently before integration. The MATLAB code used to perform this post-processing for each of the three methods is provided in shown in pseudocode.

```
Input:
- peak_locs: indices of high angular velocity peaks
- min_locs: indices of local minima in angular velocity signal
- z: summed angular velocity values per frame
- tau: threshold for meaningful motion
- min_distance: minimum spacing between valid segment points (e.g., 50 frames)

Output:
- seg_points_AngVel: finalized list of segmentation frame indices

Procedure:

1. Identify candidate segmentation points:
   raw_seg_points_AngVel ← empty list

   FOR each peak in peak_locs:
       Find the first min_loc that occurs after the peak
       IF next_min exists AND z[next_min] < tau THEN
           Append next_min to raw_seg_points_AngVel
       END IF
   END FOR

2. Filter candidates by minimum distance:
   seg_points_AngVel ← empty list
   last_added ← −∞

   FOR each point in raw_seg_points_AngVel:
       IF point − last_added ≥ min_distance THEN
           Append point to seg_points_AngVel
           last_added ← point
       END IF
   END FOR
```

Figure 3.19 MATLAB code snippet for preprocessing Method 1

```
Input:
- curvature: array of digital curvature values per frame
- L: frame offset used in curvature computation
- curv_threshold: cutoff value for sharp directional change (e.g., 0)
- min_distance: minimum spacing between segment points (e.g., 50 frames)
- num_frames: total number of motion frames

Output:
- seg_points_DirCha: finalized list of segmentation frame indices

Procedure:

1. Identify candidate segmentation points:
   raw_seg_points_DirCha ← empty list

   FOR i ← (L + 2) TO (num_frames – L – 1) DO
       IF curvature[i] is valid AND
          curvature[i] is a local minimum AND
          curvature[i] < curv_threshold THEN

           Append i to raw_seg_points_DirCha

       END IF
   END FOR

2. Filter candidates by minimum distance:
   seg_points_DirCha ← empty list
   last_added ← –∞

   FOR each point in raw_seg_points_DirCha:
       IF point – last_added ≥ min_distance THEN
           Append point to seg_points_DirCha
           last_added ← point
       END IF
   END FOR
```

Figure 3.20 MATLAB code snippet for preprocessing Method 2

```
Input:
- volume_diff: array of absolute frame-to-frame volume differences
- t: adaptive threshold for significant volume change
- min_distance: minimum spacing between segment points (e.g., 50 frames)
- num_frames: total number of frames

Output:
- seg_points_VolCha: finalized list of segmentation frame indices

Procedure:

1. Identify candidate segmentation points:
   raw_seg_points_VolCha ← empty list

   FOR i ← 2 TO num_frames − 1 DO
       IF volume_diff[i] is a local minimum AND
          volume_diff[i] < t AND
          volume_diff[i + 1] > t THEN

           Append i to raw_seg_points_VolCha

       END IF
   END FOR

2. Filter candidates by minimum distance:
   seg_points_VolCha ← empty list
   last_added ← −∞

   FOR each point in raw_seg_points_VolCha:
       IF point − last_added ≥ min_distance THEN
           Append point to seg_points_VolCha
           last_added ← point
       END IF
   END FOR
```

Figure 3.21 MATLAB code snippet for preprocessing Method 3

After obtaining the postprocessed segmentation results from each individual method, the next step involves integrating them into a unified segmentation output. Integrated segmentation candidates that fall within a 100-frame window are grouped, and the earliest frame index of each group is used as the final segmentation point. A 100-frame window was selected based on the observation that motion peaks in the motion data are separated by low-activity frames, with a minimum distance of approximately 200 frames between consecutive peaks. This ensures that each retained point represents a distinct and meaningful transition in the motion sequence. Additionally, this window also came with a purpose to maintain a final count of segmentation points that closely approximates the expected number of motion transitions, which is 11.

3.3 Evaluation Against Ground Truth Data

After integrating the three segmentation methods into a single combined result, we evaluate the accuracy of each method by comparing them to ground truth data. The four segmentation outputs, which are angular velocity, directional change, spatial volume, and the combined method, are assessed against cosine similarity metric to quantify segmentation performance.

Ground truth data is collected via a Google Form distributed to 10 experienced martial arts practitioners. Participants are shown a MATLAB-generated motion capture video of a Wushu kicking sequence, visualized as a 3D skeleton. They are instructed to identify segmentation points where the skeleton completes a kicking motion and returns to a standing stance before initiating the next kick. These standing stances are considered key transition points and correspond to local minima in the motion cycle.

Respondents are asked to write down the frame numbers corresponding to these points. All responses are aggregated, and if multiple submissions indicate frames within a 100-frame window, only the earliest frame in the group is retained as a ground truth segmentation point. This process aligns with Sonoda et al.'s postprocessing approach, which consolidates closely spaced human-annotated points into single representative events to improve consistency.

To refine this data and produce a reliable reference baseline, we adopt the filtering conditions proposed by Sonoda et al. in their segmentation study, as well as an additional condition. Specifically:

1. Redundancy Filtering (Consensus Rule)

   A frame is retained only if at least one other respondent submitted a value within a ±100 frame range. This ensures that all retained frames have some level of inter-rater agreement and eliminates outliers or noise from single-submission data.

2. Clustering and Representative Selection (Non-Maximum Suppression)

   Among closely clustered frames (within 100 frames of one another), only the earliest frame in the group is retained. This avoids overcounting and mirrors Sonoda's technique of consolidating annotations into discrete motion segments.

A 100-frame window was preferred based on the observation that peaks in the motion data are separated by low-activity frames, with a minimum distance of approximately 200 frames between consecutive peaks. The final ground truth list was used as a reference for evaluating all segmentation methods.

Once the final filtered ground truth set is established, we evaluate segmentation accuracy using the **cosine similarity metric**, which measures how well predicted segmentation points align with the expert-defined ground truth. Unlike metrics such as precision or recall that separate false positives and negatives, cosine similarity treats segmentation as a binary matching problem and quantifies alignment through a geometric formulation.

In this approach, each ground truth segmentation point is treated as a binary indicator on a timeline. A predicted point is marked as a match (1) if it falls within a tolerance window of ±100 frames of any unmatched ground truth point. Otherwise, it is considered a non-match (0). Ground truth points are also binary and fixed to 1. Each predicted point can match at most one ground truth point, and each ground truth point can only be matched once.

This formulation emphasizes both the quantity and quality of matches in a single metric. A similarity score of 1.0 means all predictions perfectly match the ground truth within the tolerance range, while lower scores indicate missed or imprecise predictions.

To implement this, we used a greedy one-to-one matching strategy. Each prediction is compared to all ground truth points, and the first unmatched ground truth point within ±100 frames is assigned to it. Once a ground truth point is matched, it is excluded from further consideration. This conservative matching prevents multiple predictions from inflating the score by aligning to the same event.

```
Input:
- predicted_methods: list of predicted segmentation points for each method
- final_gt: list of filtered ground truth segmentation points
- method_names: labels for each segmentation method
- frame_tolerance: allowable deviation between predicted and ground truth (e.g.,
100 frames)

Output:
- dot_similarities: similarity score for each method

Procedure:

Initialize:
- dot_similarities ← array of zeros, one per method

FOR each method index m:
    predicted ← predicted_methods[m]
    match_count ← 0
    matched_gt ← boolean array of same length as final_gt, all set to false

    FOR each predicted point:
        FOR each ground truth point:
            IF ground truth point not yet matched AND
               |predicted – ground_truth| ≤ frame_tolerance THEN

                match_count ← match_count + 1
                mark ground truth as matched
                BREAK inner loop

    IF match_count == 0 THEN
        similarity ← 0
    ELSE
        similarity ← match_count / (√match_count × √length(final_gt))
    END IF

    dot_similarities[m] ← similarity

    Print method name, number of predictions, match count, and similarity
END FOR
```

Figure 3.22 MATLAB code snippet to compute evaluation

Figure 3.22 shows the pseudocode for evaluating predicted points from four different methods against ground truth data.

In this study, the evaluation is implemented in MATLAB. Predicted segmentation points from each method are compared to the filtered ground truth values using the above criteria, and the resulting scores help determine which segmentation strategy most effectively approximates human-perceived transitions in the martial arts motion sequence.

This evaluation method is suitable for transition-based segmentation tasks, where the goal is to detect changes in motion phases with reasonable temporal flexibility. It avoids penalizing small time shifts while still rewarding precise event alignment.

CHAPTER IV
RESULTS AND DISCUSSIONS

This chapter presents the results of the motion segmentation analysis based on three distinct methods: angular velocity, directional change, and spatial volume variation. Each method was applied to Wushu motion capture data and evaluated individually to assess its ability to detect transitions between movement phases. Each method has its own way of thresholding and follows the criteria that was previously defined. The segmentation points obtained were visualized using MATLAB to validate their alignment with observable movement transitions. Additionally, a combined segmentation result was constructed by integrating all three methods as our final segmentation result.

To evaluate the performance of each segmentation approach, the results were compared against ground truth data collected from experienced martial arts practitioners. Quantitative metrics such as cosine similarity were calculated to measure the accuracy of each method. This chapter also provides a comparative discussion on the effectiveness, strengths, and limitations of each segmentation technique in capturing the complex, non-repetitive nature of martial arts movements.

4.1 Results

4.1.1 Segmentation by Angular Velocity

Using the segmentation method based on angular velocity and MATLAB-based computation, we successfully identified key segmentation points, which are visualized in the resulting plot.

Below, we present two plot figures illustrating the angular velocity data, with *z* plotted in blue. Figure 4.1 displays the raw angular velocity signal over time, providing an overview of the fluctuations in rotational speed throughout the Wushu kicking sequence. Figure 4.2 builds upon this by overlaying the segmentation candidate points, visualized in red stem, which are identified using our adaptive thresholding method.

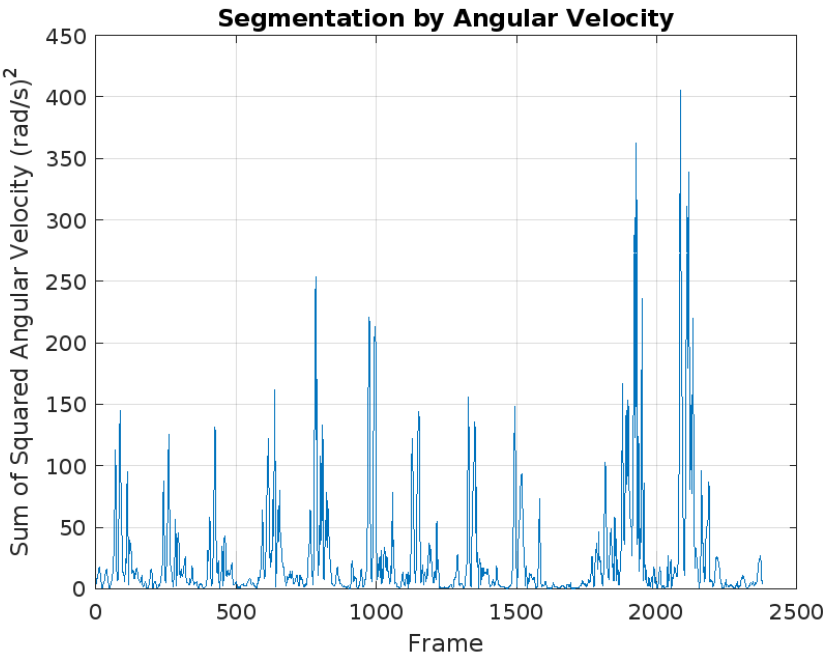


Figure 4.1 Plot of angular velocity for each frame

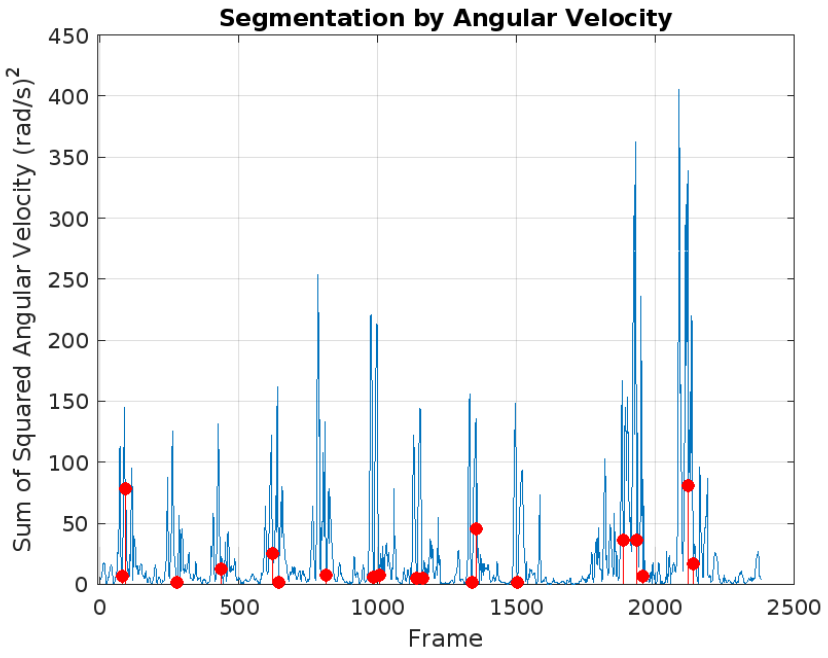


Figure 4.2 Plot of angular velocity with segmentation candidates

Based on our segmentation, a total of **19 segmentation points** were detected, indicating frames of significant change in rotational motion. The frames identified as segmentation points are as follows:

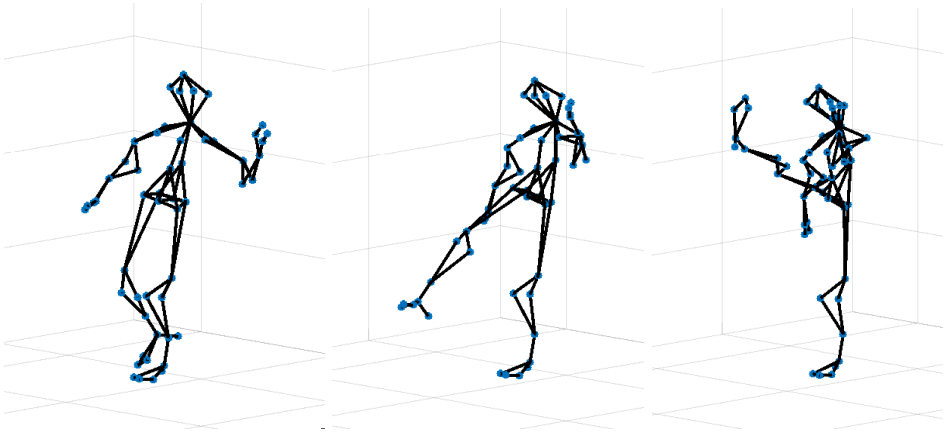
78, 90, 276, 434, 620, 642, 814, 984, 1005, 1140, 1160, 1338, 1354, 1502, 1885, 1931, 1953, 2118, 2136.

These points were extracted by analyzing the angular velocity signal across all frames. Specifically, segmentation points are determined by identifying local minima that follow local maxima exceeding a dynamic, data-driven threshold. The local maxima reflect peaks in angular velocity, typically corresponding to bursts of joint rotation, characterizing transitions between movement phases in Wushu kicking sequences.

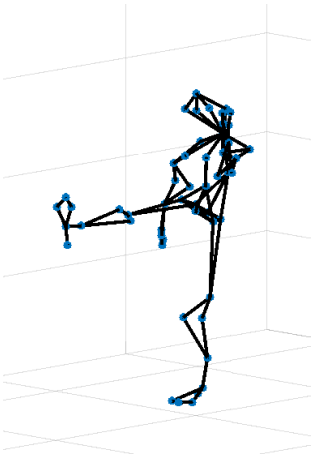
After each local maxima are detected, the algorithm traces forward in time to locate the nearest local minima. If this local minima is lower than the adaptive threshold and followed by an increase in angular velocity, it is accepted as a valid segmentation point. This approach ensures that segmentation occurs at natural transition points in the motion cycle, such as the return to a neutral or standing stance following a kick.

The effectiveness of this method is visually supported by the segmentation plot, where most detected points closely correspond to observable transitions in the motion trajectory. However, some frames, such as 90, 1354, and 2118, exhibited values too high to reflect actual motion pauses, potentially leading to false positives. Additionally, certain frames such as 78 and 1885 revealed premature dips or peaks that occurred before the actual transition, further indicating noise or minor fluctuations in the angular velocity signal. As such, the true validity of the segmentation results will be assessed through comparison with manually annotated ground truth data in a subsequent section.

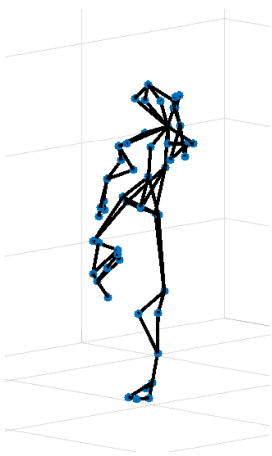
Figure 4.3 shows the 19 skeleton frames according to angular velocity.



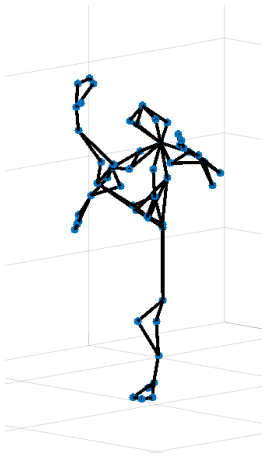
(78) (90) (276)



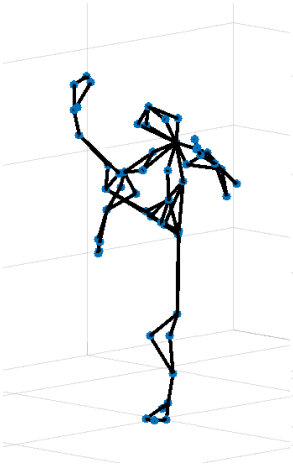
(434)



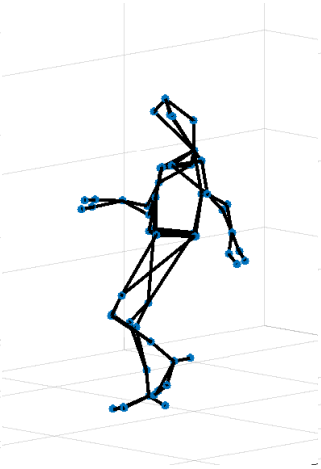
(620)



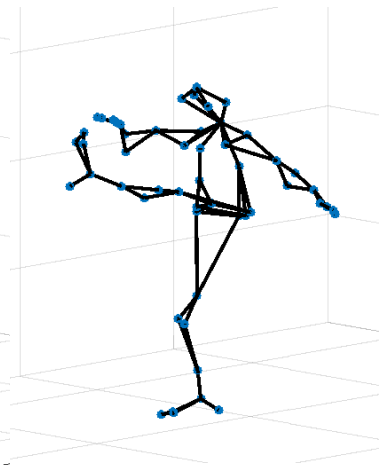
(642)



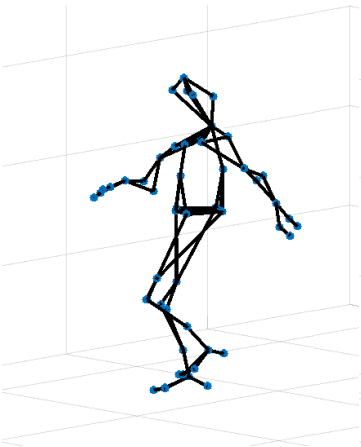
(814)



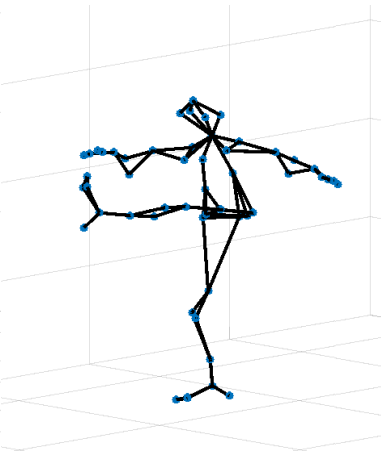
(984)



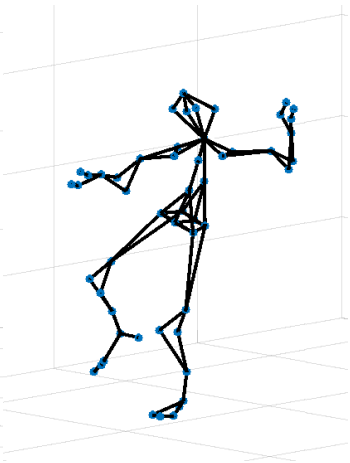
(1005)



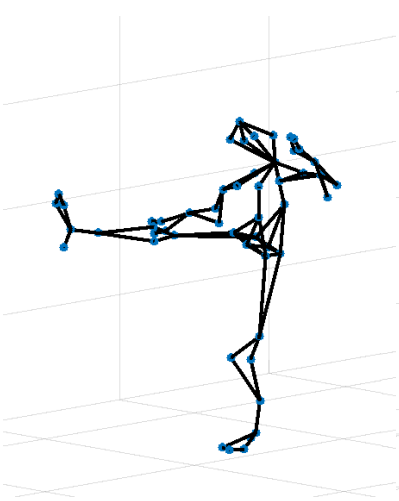
(1140)



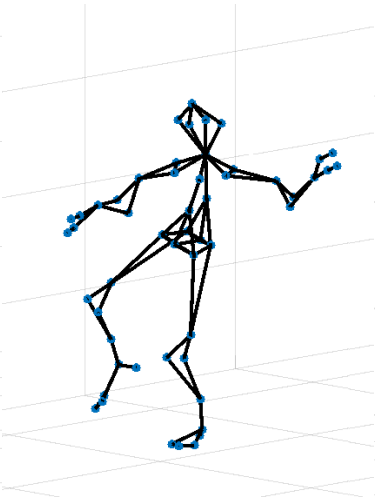
(1160)



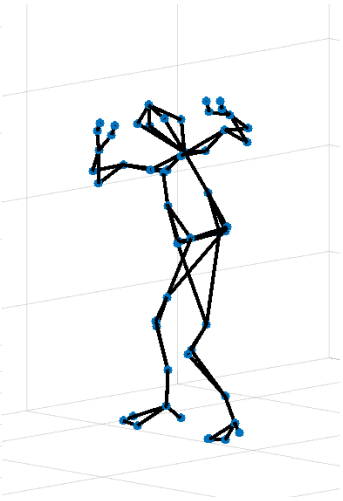
(1338)



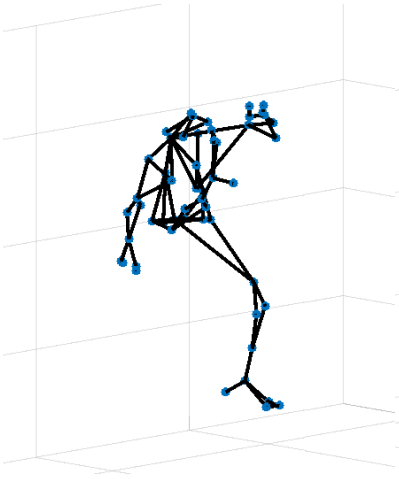
(1354)



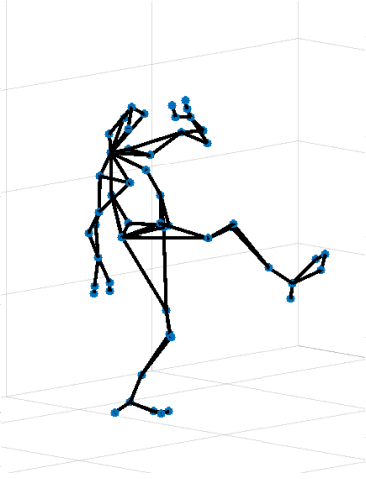
(1502)



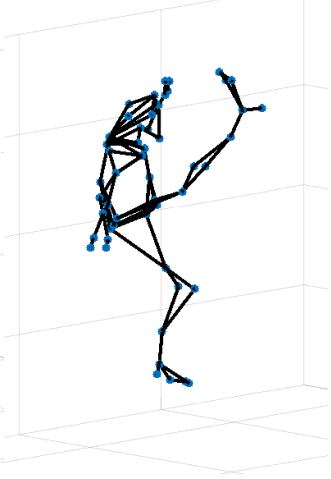
(1885)



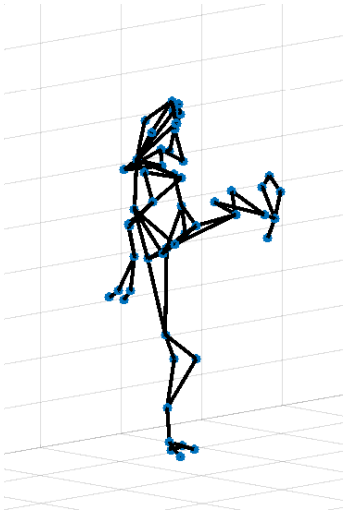
(1931)



(1953)



(2118)



(2136)

Figure 4.3 19 segmented frames according to angular velocity

4.1.2 Segmentation by Change of Direction

In the context of segmentation based on directional change using digital curvature, we successfully identified key segmentation points, which are visualized in the corresponding plots.

Following the method proposed by Sonoda et al., we applied a minimum displacement threshold of 30 mm to exclude frames with negligible movement. This threshold ensures that digital curvature is only computed when both directional vectors represent meaningful displacement, effectively filtering out static or low-motion frames that do not signify true directional changes.

To illustrate the impact of thresholding, we present two comparative segmentation plots, as well as a regular plot without implementing segmentation. Figure 4.4 displays the digital curvature plot without segmentation, whereas Figure 4.5 displays the digital curvature segmentation plots generated using two different conditions: without thresholding (0 mm) and with a 30 mm displacement threshold.

For Figure 4.5, in the 0 mm threshold condition (a), the digital curvature values are computed at every frame, resulting in a continuous plot. However, this approach leads to over-segmentation, as minor jitters or noise in the marker trajectory can produce curvature spikes that are falsely interpreted as significant motion changes. The inclusion of such insignificant points compromises segmentation quality.

In contrast, the 30 mm threshold condition (b) demonstrates the benefit of filtering. By excluding frames where either vector is below the displacement threshold, the method reduces noise and improves segmentation precision, retaining only frames associated with substantial directional transitions.

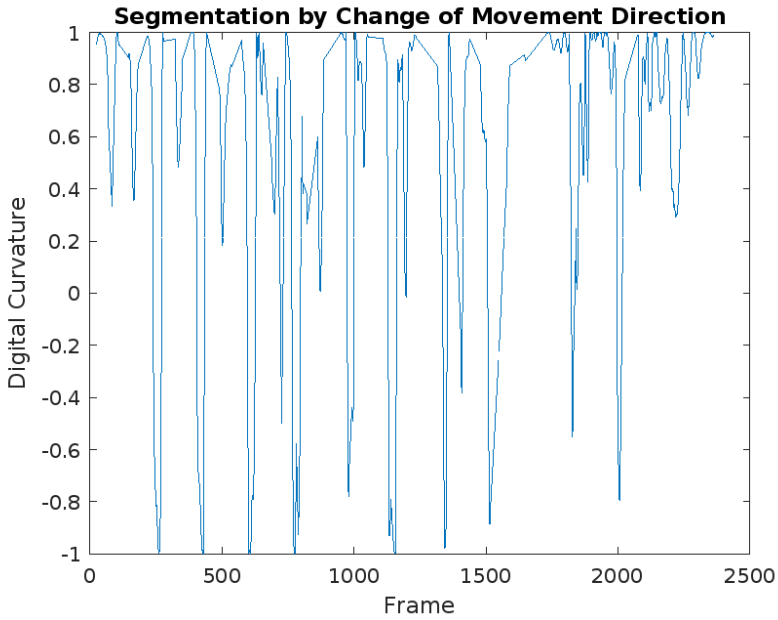
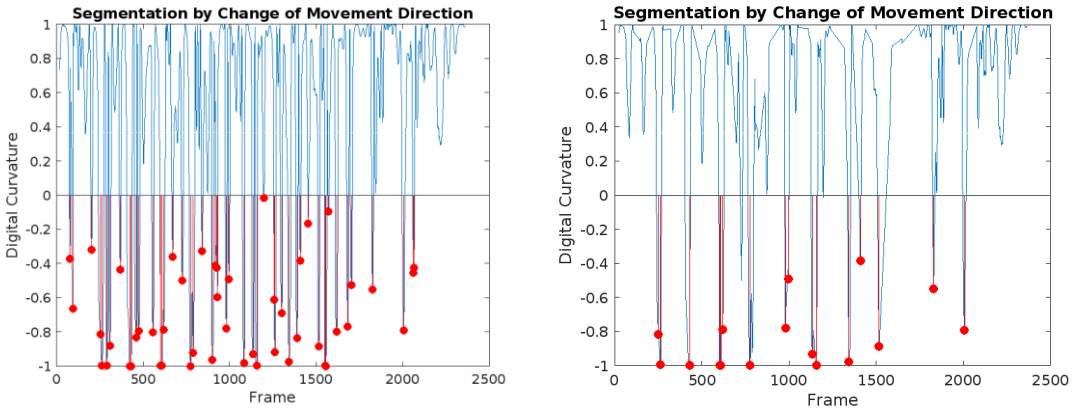


Figure 4.4 Plot of digital curvature for each frame



(a) (b)

Figure 4.5 Plot of digital curvature with segmentation candidates

In addition to the digital curvature plots, we present another two comparative figures and one regular figure, illustrating the trajectory of the root marker (LMWT) in the horizontal plane. Figure 4.6 shows the Wushu root marker trajectory projected onto the XY plane without implementing segmentation, and Figure 4.7 shows the Wushu root marker trajectory segmentation generated using two different conditions: without thresholding (0 mm) (a) and with a 30 mm (b) displacement threshold, allowing visual interpretation of directional changes during the kicking sequence.

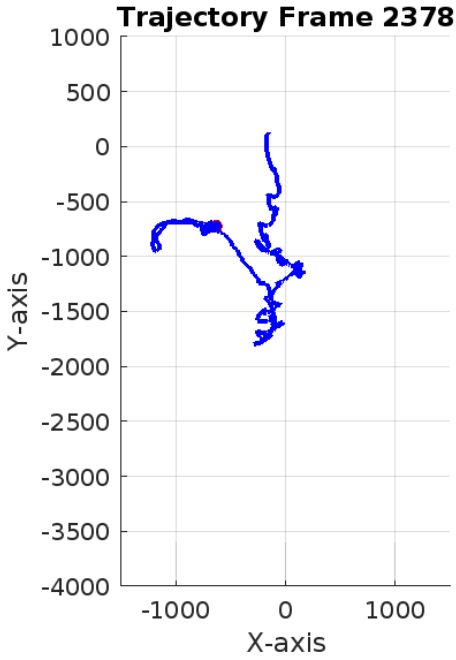
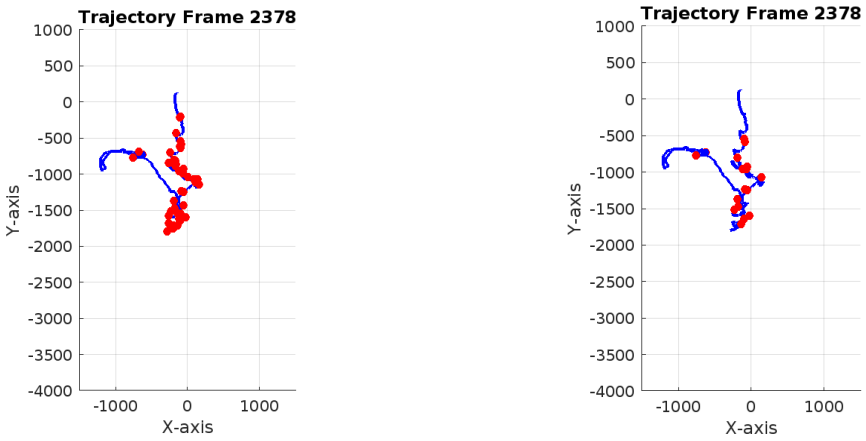


Figure 4.6 Plot of Wushu kicking motion trajectory in an XY-horizontal plane



(a) (b)

Figure 4.7 Plot of Wushu kicking motion trajectory with segmentation candidates

From the raw digital curvature data, a total of 49 segmentation candidate frames were initially detected prior to applying any threshold. These frames are:

77, 93, 201, 252, 262, 289, 307, 368, 422, 430, 460, 473, 557, 602, 608, 618, 671, 727, 775, 788, 841, 900, 918, 927, 930, 981, 996, 1083, 1134, 1156, 1198, 1256, 1262, 1302, 1344, 1389, 1409, 1453, 1515, 1551, 1555, 1572, 1618, 1683, 1705, 1827, 2006, 2063, 2065.

These excessive candidates were primarily caused by small positional fluctuations that produced abrupt spikes in curvature despite lacking meaningful directional change.

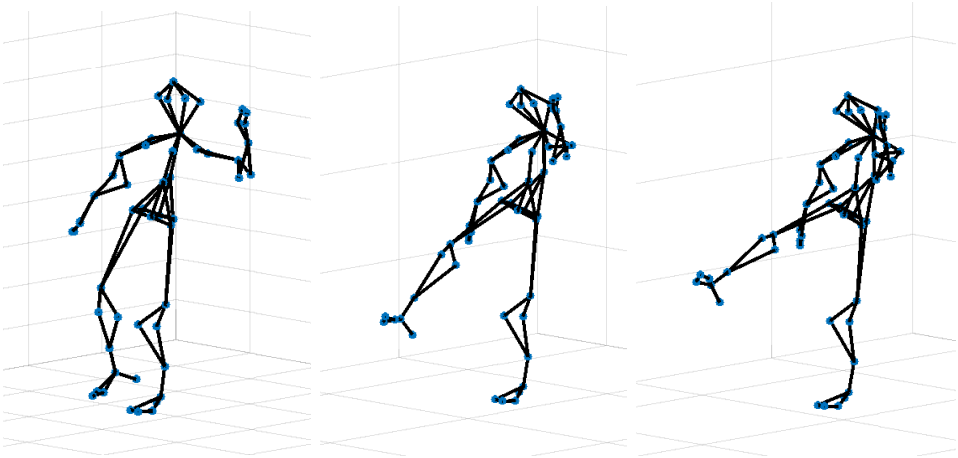
As a result of this filtering, the number of **segmentation points** was reduced to **16**, effectively removing noise and retaining only significant directional transitions. The filtered segmentation frames are:

252, 262, 430, 602, 608, 618, 775, 981, 996, 1134, 1156, 1344, 1409, 1515, 1827, 2006.

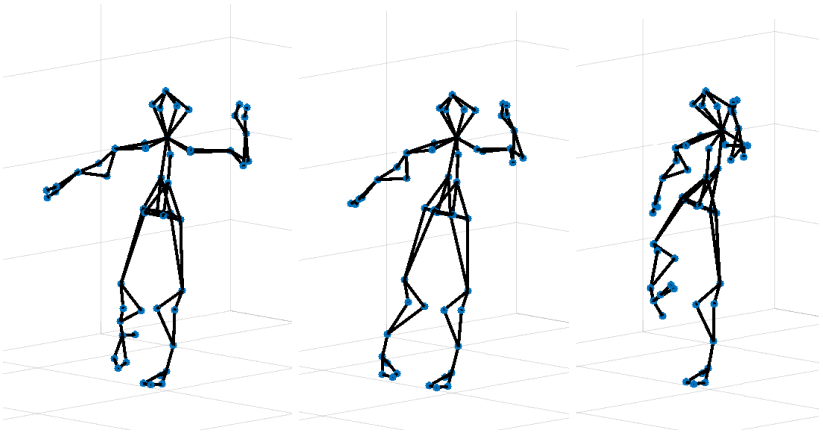
In this method, segmentation candidates are identified when the root marker (LMWT) exhibits a trajectory shift of approximately 90 degrees, or when the digital curvature value crosses zero and approaches -1, indicating a sharp and meaningful change in movement direction.

However, the effectiveness of this method can be questioned from a visual standpoint. In the digital curvature plot, segmentation points that exceed the threshold and surpass a value of 0 are detected accurately. However, in the trajectory plot, it is evident that some frames exhibiting sharp directional changes were not identified by the computation. This makes it difficult to visually confirm whether the method has reliably captured segmentation points based on directional change. Therefore, the true validity of the results will be assessed through comparison with ground truth data in a later section.

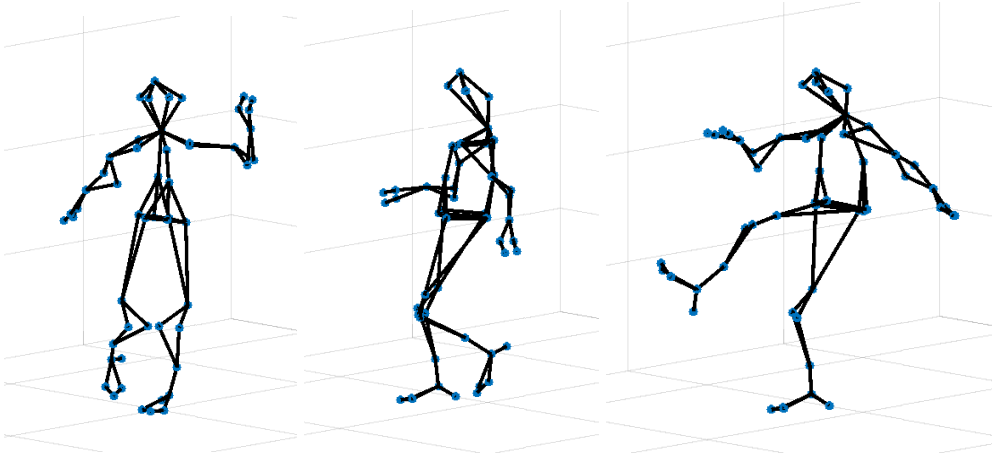
Figure 4.8 illustrates the 16 skeleton frames identified based on directional change using the digital curvature method, after applying a 30 mm displacement threshold.



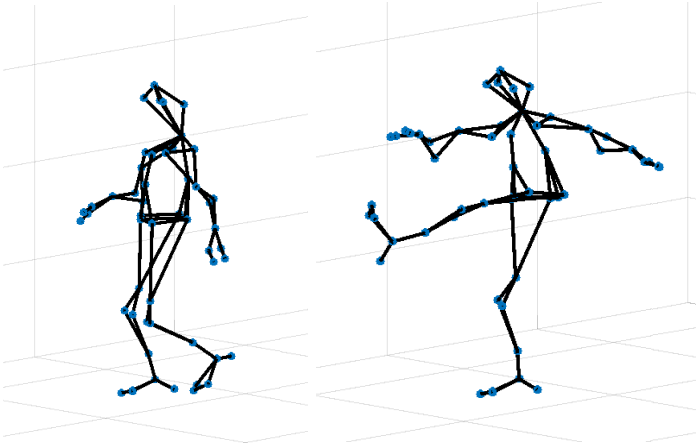
(252) (262) (430)



(602) (608) (618)



(775) (981) (996)



(1134) (1156)

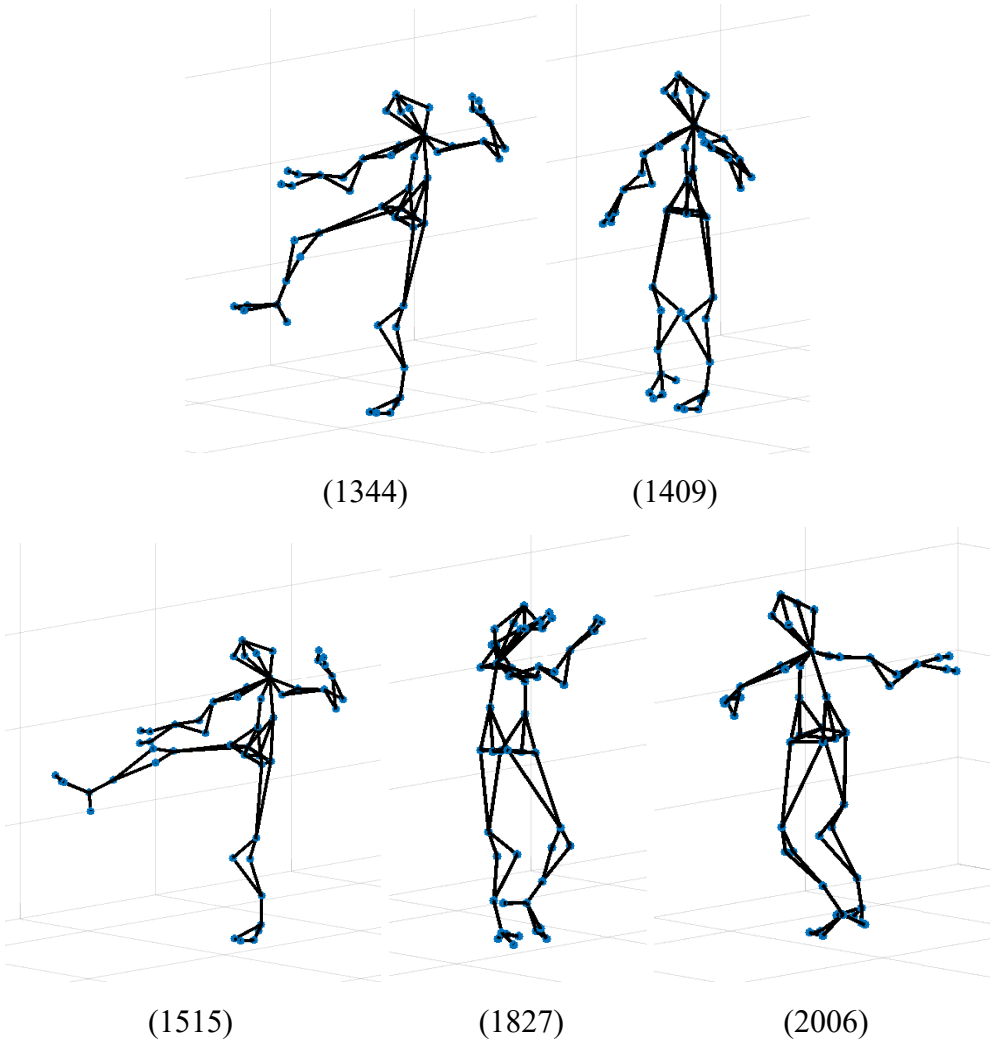


Figure 4.8 16 segmented frames according to directional change in digital curvature

4.1.3 Segmentation by Change of Volume

Lastly, we have also succesfully compute segmentation points based on the spatial volume created by the body via convex hull, which are visualized in the plot below.

We present two plot figures illustrating the volume difference data or D_i plotted in blue. Figure 4.9 displays the raw volume difference signal over time, providing an overview of the fluctuations in spatial volume throughout the Wushu kicking sequence. Figure 4.10 builds upon this by overlaying the segmentation candidate points, visualized in red stem, which are identified using our adaptive thresholding method.

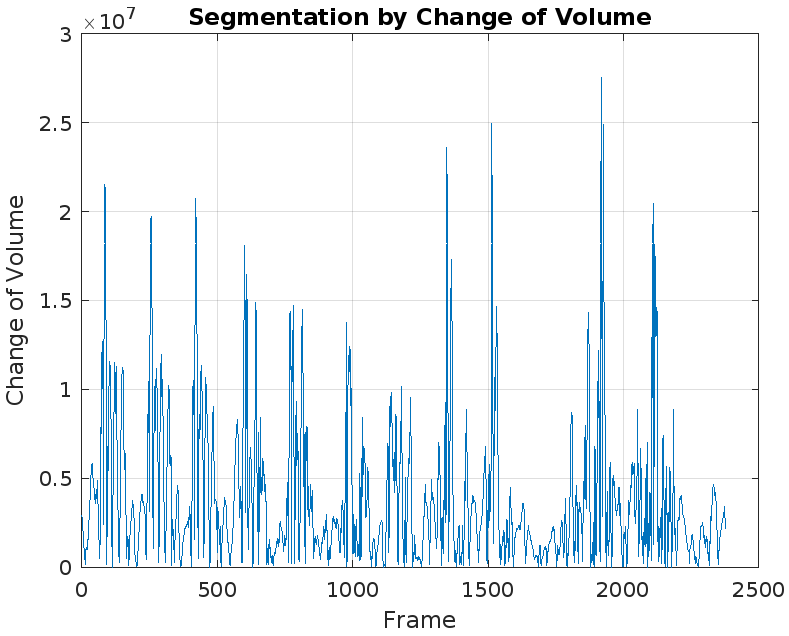


Figure 4.9 Plot of volume difference for each frame

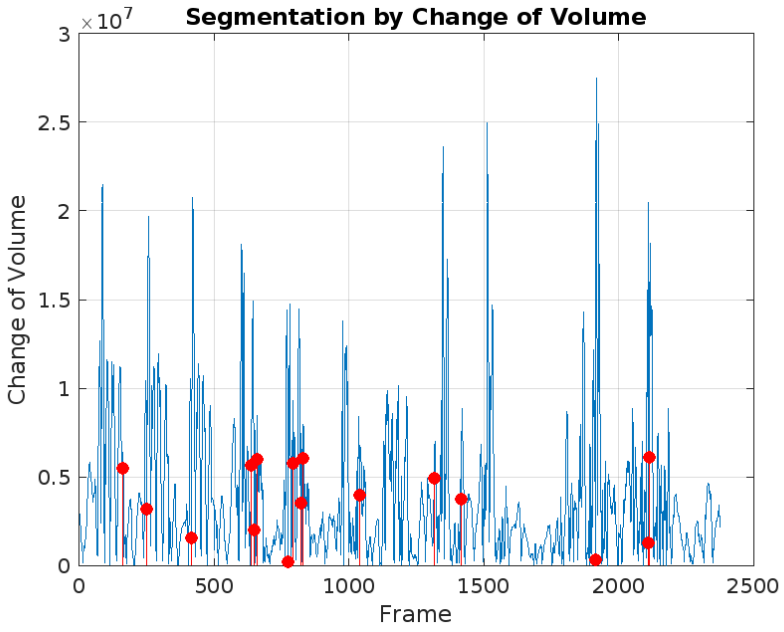


Figure 4.10 Plot of volume difference with segmentation candidates

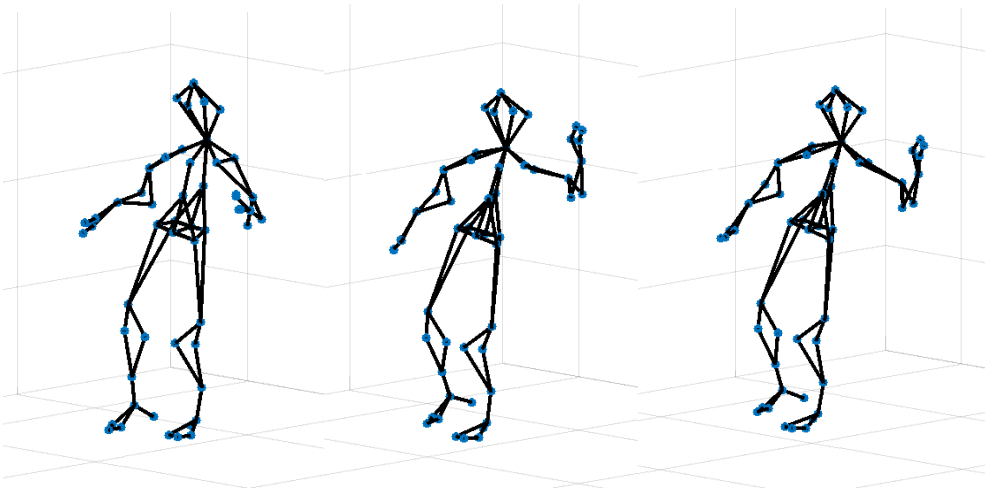
Based on our segmentation, a total of **16 segmentation points** were detected, indicating frames of significant change in spatial motion. The frames identified as segmentation points are as follows:

160, 251, 416, 638, 650, 660, 775, 794, 821, 831, 1038, 1317, 1416, 1914, 2112, 2114.

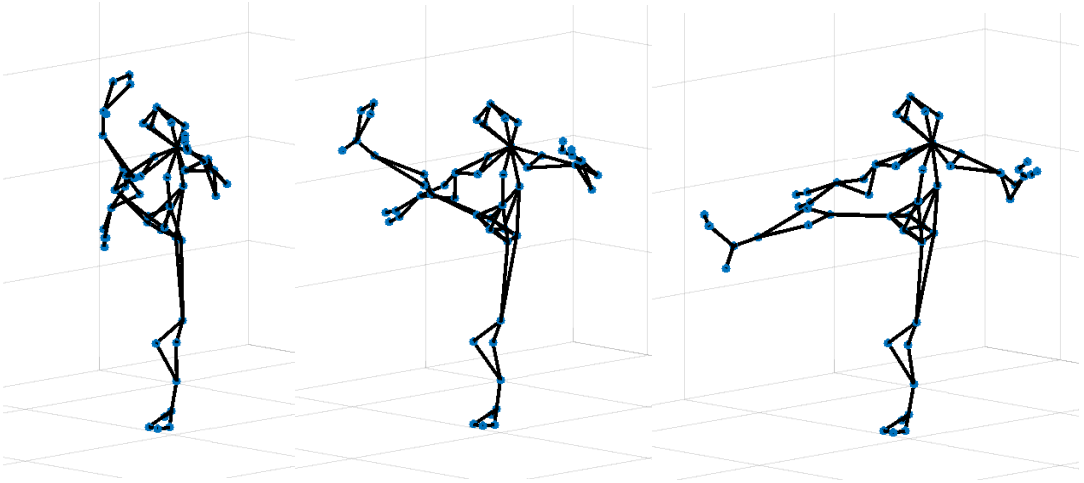
In this method, segmentation candidates are identified when the volume difference signal reaches a local minimum, reflecting a brief pause in the body's spatial posture. These minima often occur between phases of expansion and contraction during dynamic motion. To ensure that only significant postural changes are considered, an adaptive threshold t is applied to filter out minor fluctuations. Local minima are selected as segmentation points only if they fall below this threshold, which is calculated as $t = 0.23 \cdot \text{range}(D)$, where D is the set of frame-to-frame convex hull volume differences. To maintain robustness across varying motion intensities, a minimum fallback threshold is also enforced: $t = \max(0.23 \cdot \text{range}(D), 2\times10^6)$. This ensures that only dips following large-scale movement bursts are considered valid.

However, the effectiveness of this method can be critically assessed from a visual standpoint. In the volume difference plot, several inconsistencies are apparent in the segmentation results. Notably, some prominent peaks that likely indicate meaningful transitions are not detected, such as those occurring between frames 1100-1300 and 1500-1800. Additionally, certain segmentation points are clustered too closely together, occasionally originating from the same peak area, such as frames 638, 650, and 660. These observations raise concerns about the method's reliability in capturing meaningful segmentation points based on spatial or volumetric change, particularly in the context of Wushu kicking motion. Therefore, the validity of the segmentation outcomes will be further evaluated through comparison with ground truth annotations in a later section.

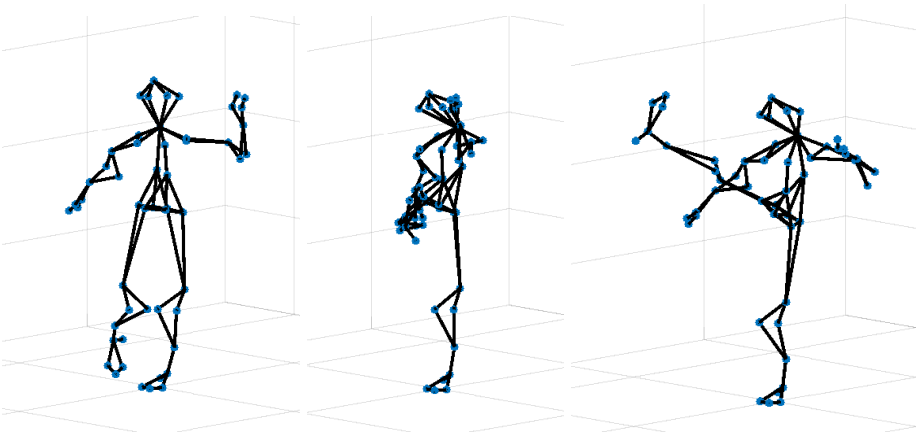
Figure 4.11 shows the 16 skeleton frames according to change of volume.



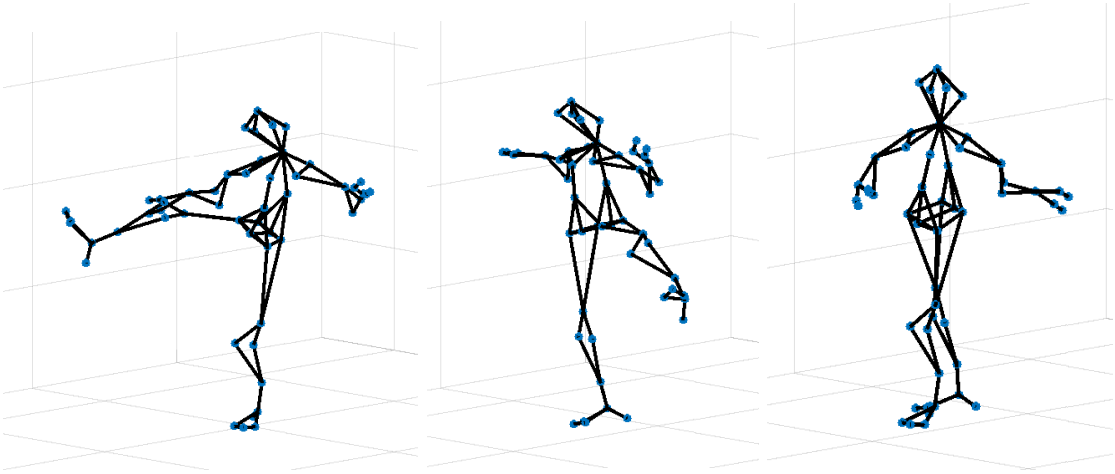
(160) (251) (416)



(638) (650) (660)



(775) (794) (821)



(831) (1038) (1317)

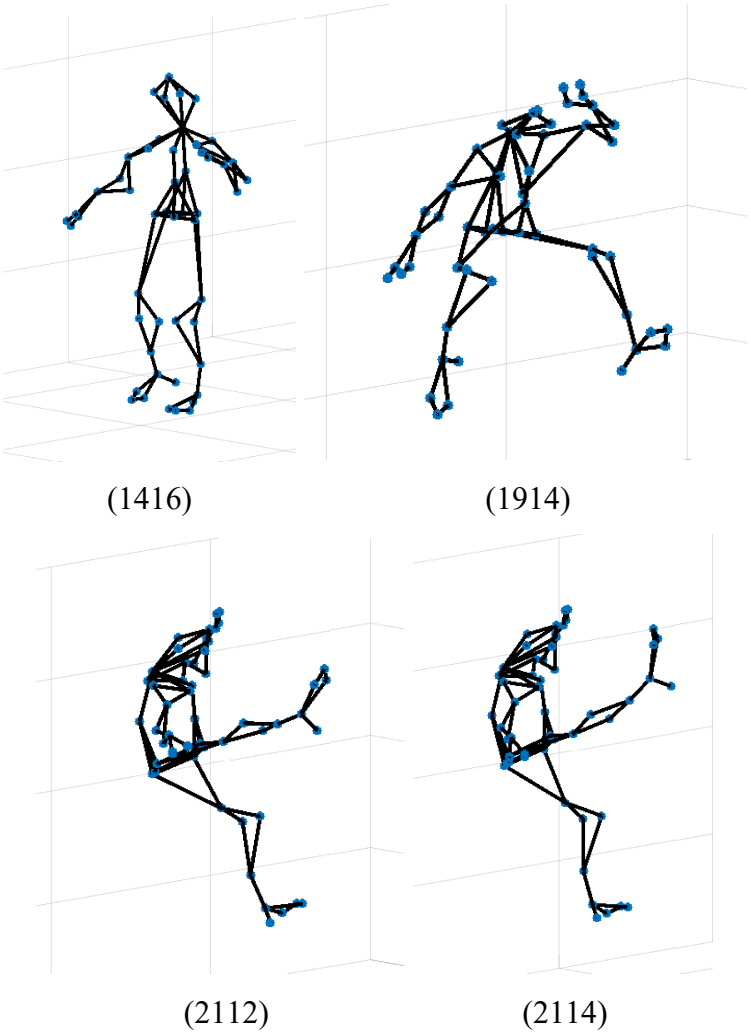


Figure 4.11 16 segmented frames according to volume change

4.1.4 Integrating Segmentation Method Results

After identifying the raw segmentation candidates from each of the three methods, we proceed to integrate their results into a combined segmentation output. However, before integration, a postprocessing step is applied to each method individually. In this step, segmentation points that occur within a 50-frame window are grouped, and only the earliest frame in each group is retained. This approach helps eliminate redundant detections and ensures that each segmentation point represents a distinct transition.

The results of this postprocessing step for each method are visualized in the following figures. Figure 4.12, Figure 4.13, and Figure 4.14 present the segmentation plots for the angular velocity method, the direction change method using digital curvature, and the volume change method using volume difference, respectively.

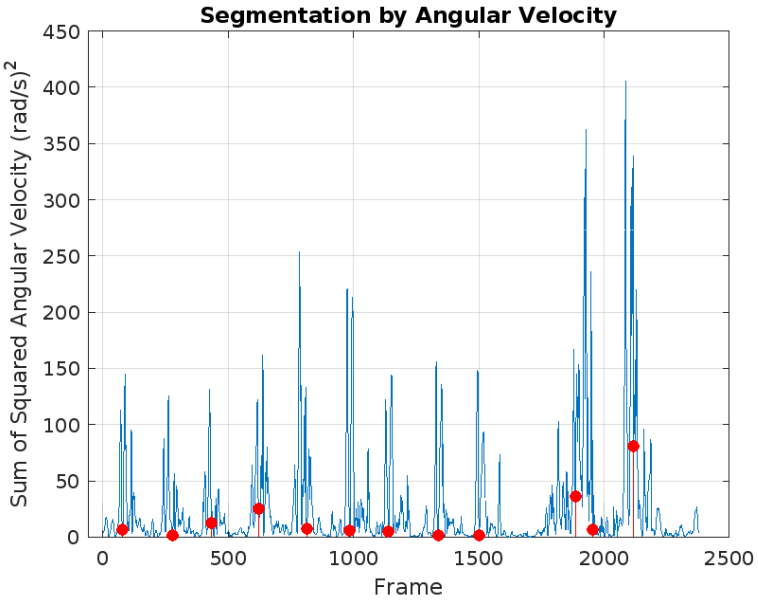


Figure 4.12 Plot of angular velocity after postprocessing

Figure 4.12 shows the angular velocity segmentation result after postprocessing. The number of candidate frames was reduced from **19** to **12**. The remaining frames are:

78, 276, 434, 620, 814, 984, 1140, 1338, 1502, 1885, 1953, 2118.

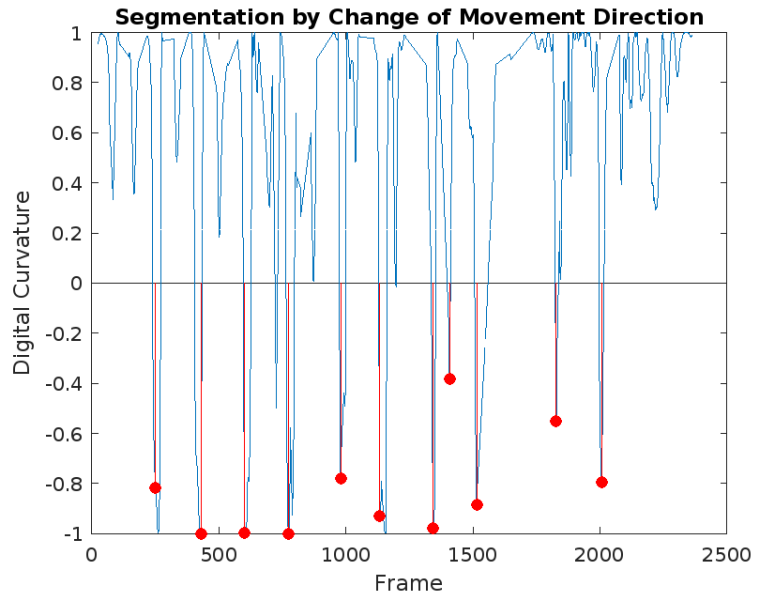


Figure 4.13 Plot of digital curvature after postprocessing

Figure 4.13 presents the direction change method after postprocessing. The segmentation points were reduced from **16** to **11**, with the remaining frames:

252, 430, 602, 775, 981, 1134, 1344, 1409, 1515, 1827, 2006.

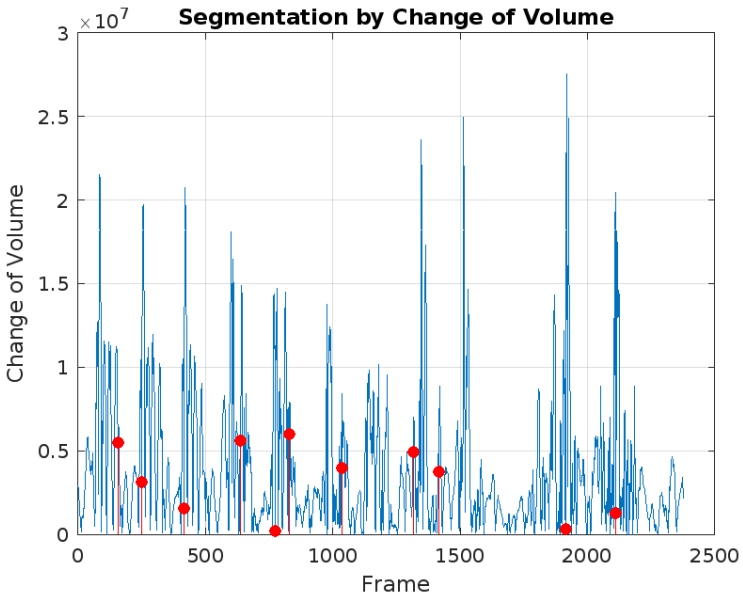


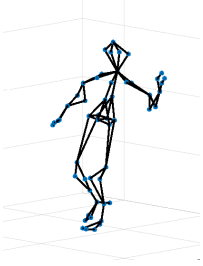
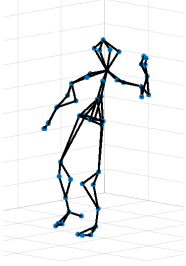
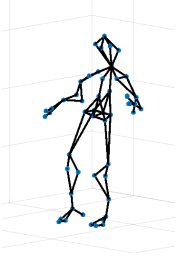
Figure 4.14 Plot of volume difference after postprocessing

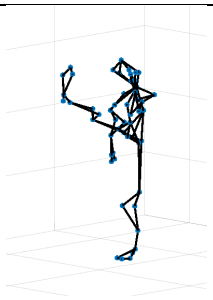
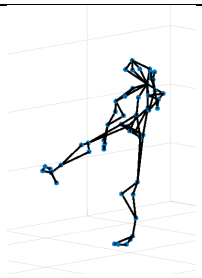
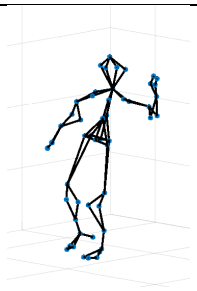
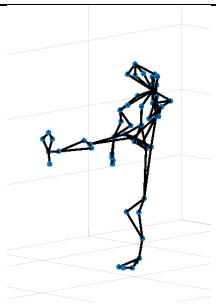
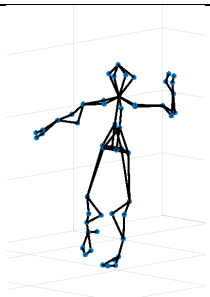
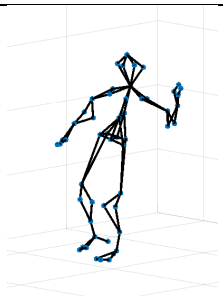
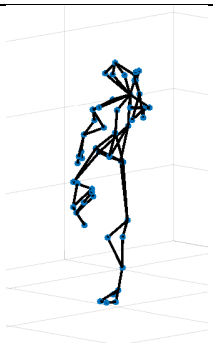
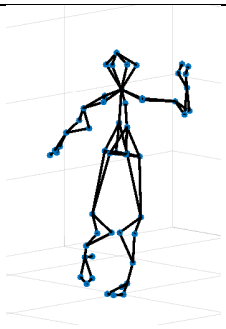
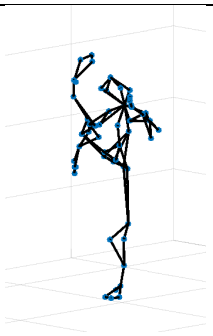
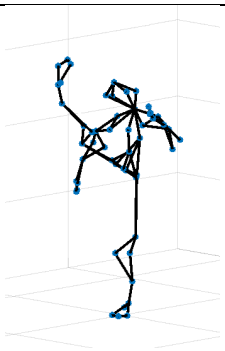
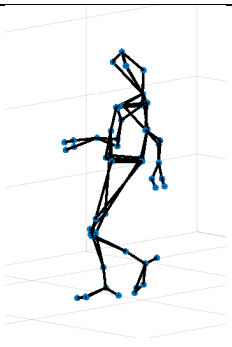
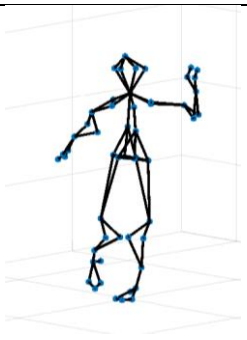
Figure 4.14 displays the results of the volume change method, where the candidates were also reduced from **16** to **11**. The retained segmentation frames are:

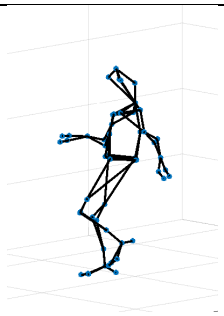
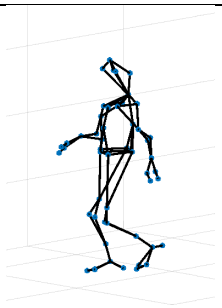
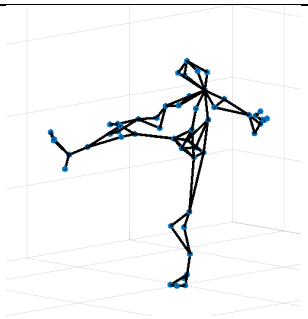
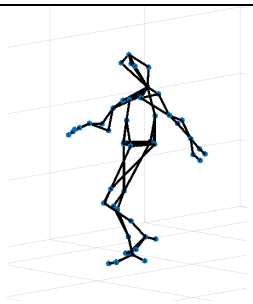
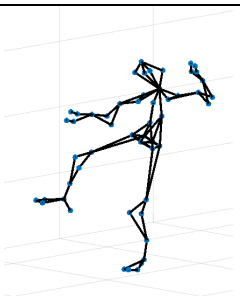
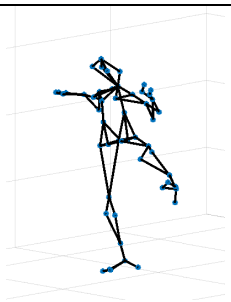
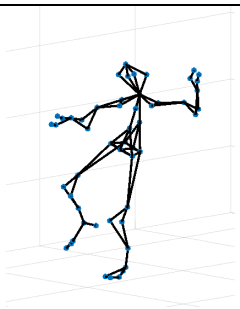
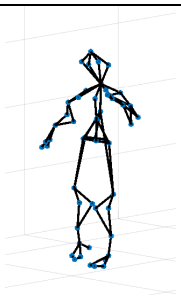
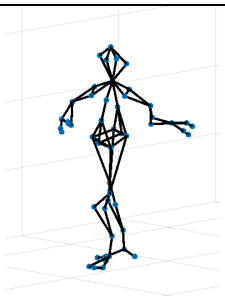
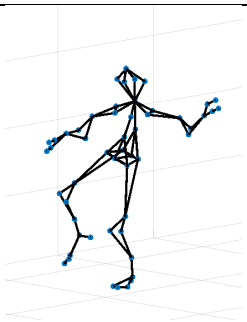
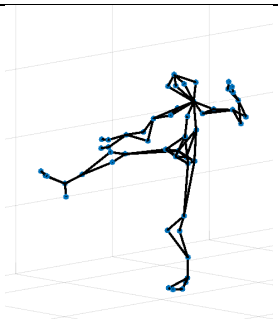
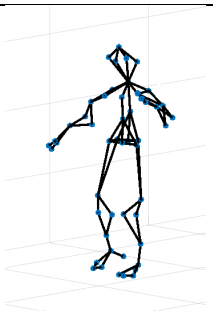
160, 251, 416, 638, 775, 831, 1038, 1317, 1416, 1914, 2112.

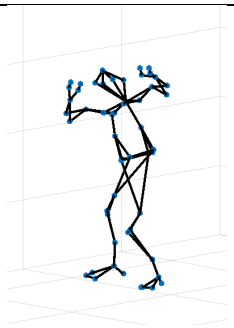
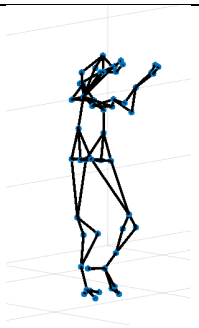
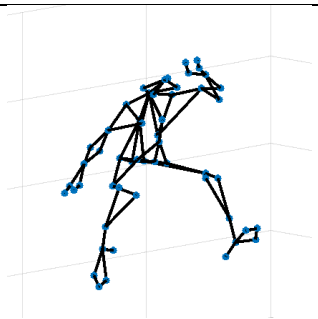
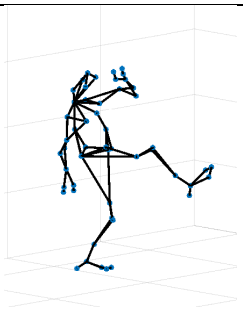
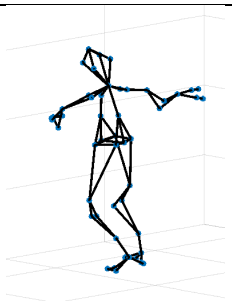
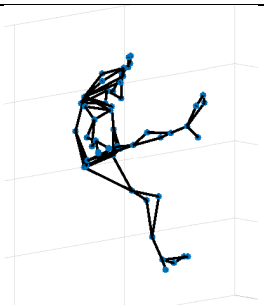
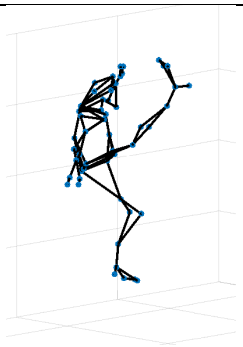
The integration of the three segmentation methods resulted in a total of **33** unique segmentation candidates. Although the combined output initially contained **34** entries, one overlapping frame, which is frame 775, was produced by both directional change and volume change method and therefore counted only once. A summary of the segmentation frames, including their visualizations for each method after post-processing, is presented in Table 4.1.

Table 4.1 Skeleton frame candidates from each three method after postprocessing

No.	Angular Velocity	Directional Change	Volume Change
1			

No.	Angular Velocity	Directional Change	Volume Change
	78	252	160
2	 276	 430	 251
3	 434	 602	 416
4	 620	 775	 638
5	 814	 981	 775

No.	Angular Velocity	Directional Change	Volume Change
6	 984	 1134	 831
7	 1140	 1344	 1038
8	 1338	 1409	 1317
9	 1502	 1515	 1416

No.	Angular Velocity	Directional Change	Volume Change
10	 1885	 1827	 1914
11	 1953	 2006	 2112
12	 2118		

From the integrated results of the three segmentation methods, several frames were found to be temporally close to one another. To address this redundancy, an additional grouping step was applied: frames occurring within a 100-frame window (100 frames before, 100 frames ahead) were grouped together, and only the earliest frame in each group was retained as the representative. This post-processing step ensures that the final segmentation output from the integration method contains only distinct and meaningful candidates, eliminating closely spaced duplicates while preserving key transition points.

The segmentation results for all three methods are visualized in the raster plot in Figure 4.15. The bottom row of the plot illustrates the integration of all methods, providing a clear overview of the combined segmentation points.

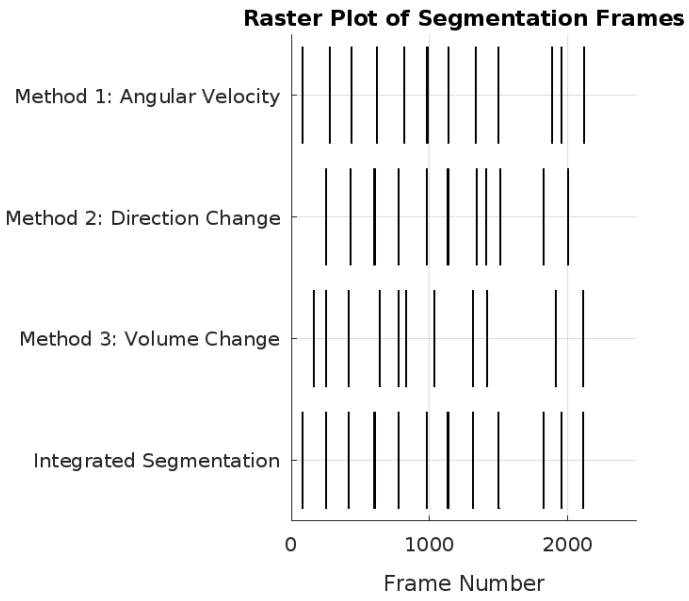
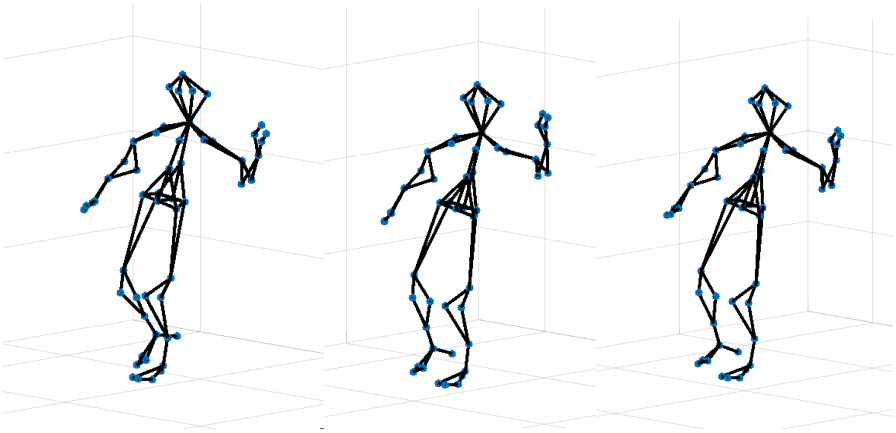


Figure 4.15 Raster plot showing the integrated and postprocessed result

After integration and postprocessing, the final segmentation result for the combined method consists of **12** unique segmentation candidates. These frames are as follows:

78, 251, 416, 602, 775, 981, 1134, 1317, 1502, 1827, 1953, 2112.

Figure 4.16 shows the 12 final integrated skeleton frames.



(78) (251) (416)

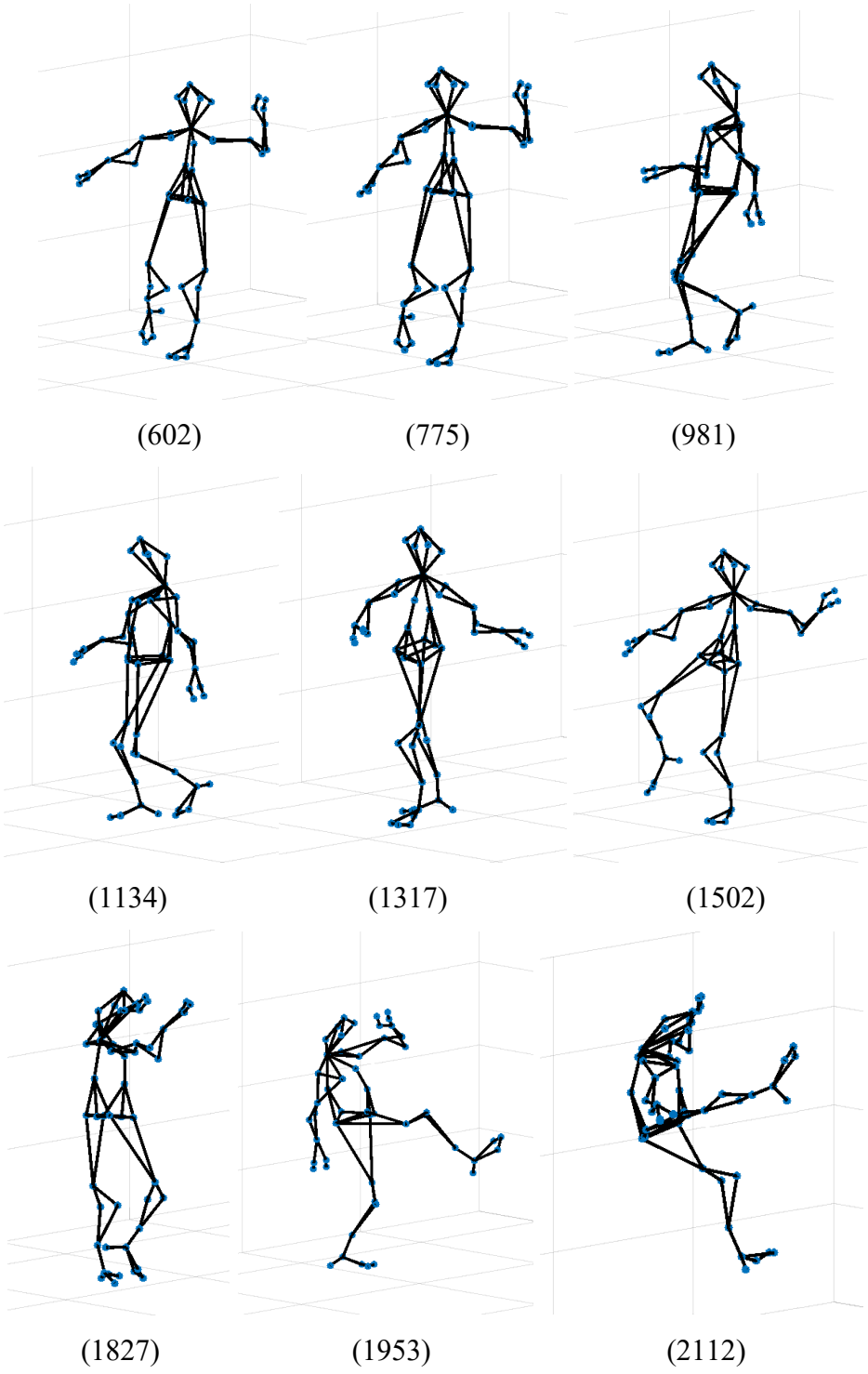


Figure 4.16 12 segmented frames according to integration

4.1.5 Evaluation Against Ground Truth Data

To evaluate segmentation accuracy, four methods, which are angular velocity, directional change, volume change, and the integrated result, were tested against manually labeled ground truth data. These ground truth points were gathered from 10 expert respondents via a Google

Form, where each participant marked frame numbers indicating transitions (i.e., return to a standing stance) in a 3D motion capture video of a Wushu kick.

Prior to any filtering or consensus processing, the raw ground truth dataset consisted of **103** distinct frame annotations provided by the respondents. These frames reflect individual perceptions of segmentation points based on the visualized motion sequence, and include overlapping, isolated, and closely spaced inputs that require further refinement to ensure consistency and reliability in evaluation.

To ensure reliability, the raw annotations were filtered using Sonoda et al.'s method. This included two steps:

(1) Redundancy filtering, which retains only frames confirmed by at least one other respondent within a ±100 frame range, and

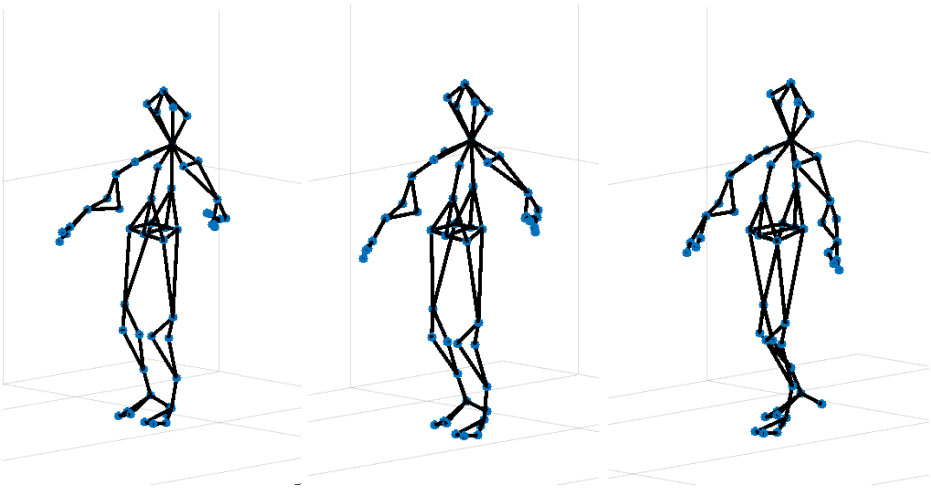
(2) Non-maximum suppression, where only the earliest frame from clusters of closely spaced annotations is kept.

The first result was 12 validated ground truth points used as reference. The frames are:

169, 342, 522, 705, 880, 1070, 1226, 1438, 1595, 1992, 2162, 2265.

To evaluate the segmentation results using cosine similarity metric, we adopted the same evaluation strategy used by Sonoda et al., which employs a *greedy matching approach*.

For reference, Figure 4.17 shows the 12 skeleton frames according to ground truth data, and Figure 4.18 shows the raster plot, comparing the four methods against ground truth data.



(169) (342) (522)

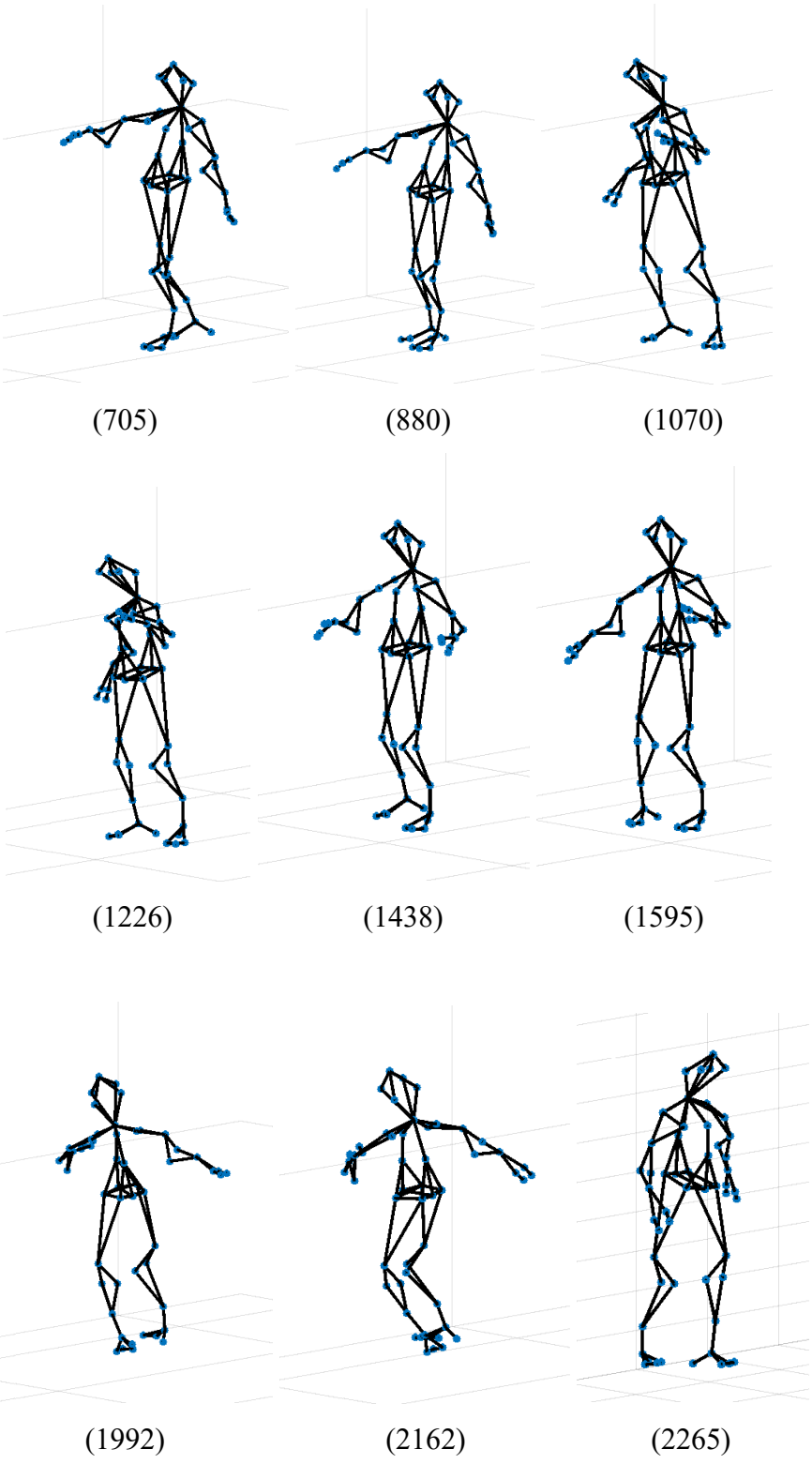


Figure 4.17 12 segmented frames according to ground truth data

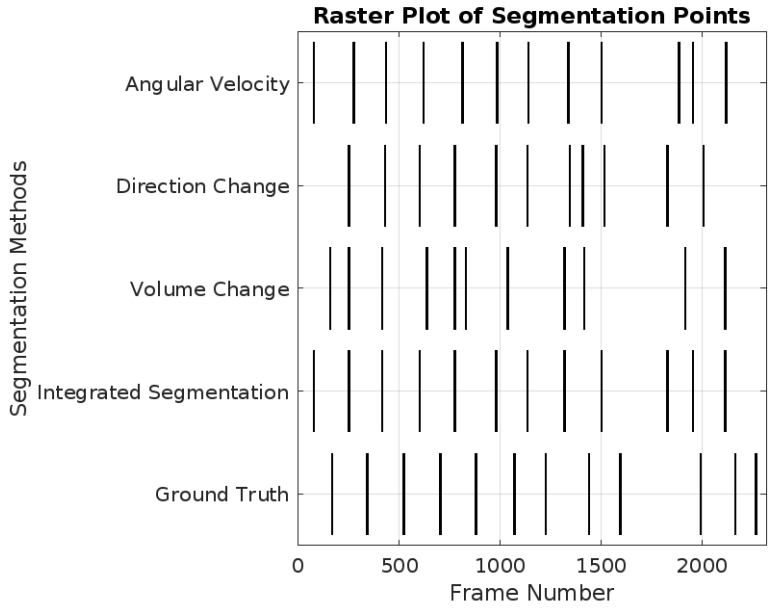


Figure 4.18 Raster plot comparing 4 methods against ground truth data

Table 4.2 presents the evaluation results for each segmentation method against 12 frames of ground truth, using a ±100 frame tolerance window, and Figure 4.19 presents the evaluation results as a grouped bar chart.

Table 4.2 Result of evaluation metrics

Method	**Total Predictions (A)**	**Dot Product Predictions ($A \cdot B$)**	**Similarity ($\frac{A \cdot B}{\|A\| \cdot \|B\|}$)**
Angular Velocity	12	11	0.9574
Direction Change	11	9	0.8660
Volume Change	11	9	0.8660
Integrated	12	9	0.8660

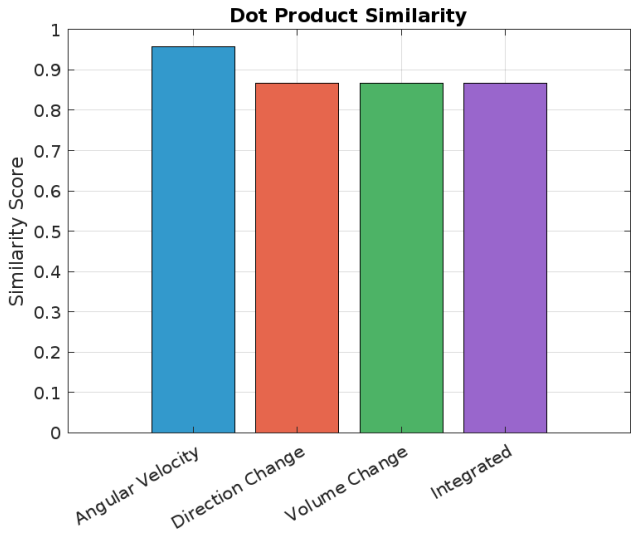


Figure 4.19 Evaluation results visualized in grouped bar chart

4.2 Discussions

4.2.1 Performance Comparison of Segmentation Methods

This study evaluated four segmentation methods: angular velocity, directional change, volume change, and a combination of the three. Their effectiveness was measured using cosine similarity against a filtered ground truth set of known motion transitions.

The angular velocity method achieved the highest score, with a similarity of 0.9574, correctly matching 11 out of 12 transitions. This suggests that sharp changes in joint rotation are strong and consistent indicators of phase transitions in Wushu kicks. Kicking actions often involve bursts of angular movement, especially around the hip and knee joints, which makes this method effective. Its high alignment with ground truth shows that it can detect key turning points in the motion cycle while minimizing false matches.

The volume change method, based on fluctuations in the convex hull volume of the body, achieved a similarity of 0.8660. It matched 9 ground truth transitions. This method works by capturing the difference between compact and extended postures. Since Wushu kicks naturally alternate between these two states, the volume cue tends to track motion phases well. However, its performance may be limited in cases where posture change is subtle or masked by arm movements that don't reflect a full-body transition, leading to a few missed matches.

The direction change method, which also scored 0.8660, detects motion shifts based on trajectory curvature, particularly of the root marker projected onto the horizontal plane. It performed worse than expected, missing 3 transitions. This is likely because direction changes in footwork can be small or smoothed out during fast or circular motions, especially if the root

marker doesn't sharply pivot. Additionally, some transitions occur in place (e.g. a kick from a stationary position), meaning direction may not noticeably change even when the phase does. This explains its lower match count compared to angular velocity.

Despite having the same similarity score, the direction change and volume change methods failed to match different subsets of the ground truth transitions, revealing distinct weaknesses in each approach. The direction change method missed several key transitions, including those near 880, 2162, and 2265. These missed points often occurred during stationary kicks or when the root trajectory lacked a sharp turn, which this method depends on. This suggests that direction-based segmentation struggles when movement happens in place or along smooth arcs without clear directional shifts. On the other hand, the volume change method failed to capture transitions near 522, 1595, and 2265, which are moments that featured a clear change in trajectory but did not involve a large shift in body posture. In these cases, the spatial structure of the body remained relatively stable, even as the motion phase changed. These differences highlight that while both methods contribute valuable cues, their limitations emerge under different movement conditions. Each method overlooks specific types of transitions, suggesting that future segmentation strategies should treat them as complementary rather than interchangeable.

4.2.2 Effectiveness of the Integrated Method

The integrated method merges segmentation outputs from all three feature-based techniques using a 100-frame tolerance window. Although its similarity score (0.8660) matched that of the direction and volume change methods, the specific ground truth points it matched were different. It successfully identified 9 transitions, but missed key points at frames 880, 1595, and 2265. This suggests that while integration helps cover more ground, a simple union of results does not necessarily improve overall accuracy. Some transitions detected by the integrated method were not matched by any individual method, indicating that combining features can recover overlooked phases. However, the integration also passed through false positives, particularly from weaker individual signals, which limited its overall gain. These findings imply that integration needs to be more selective, using strategies such as confidence scoring, weighting based on method reliability, or post-filtering, to prevent noise from diluting the segmentation output.

4.2.3 Interpretation of Findings

The results clearly show that angular velocity is the most accurate and consistent method for segmenting Wushu kicking motions in this dataset. Its alignment with expert-labeled transitions shows that rotational bursts offer a strong signal for detecting phase changes.

The fact that volume change and direction change achieved the same similarity score but matched different sets of ground truth transitions highlights their distinct roles. Volume change is more sensitive to spatial expansion and posture, while direction change responds to orientation shifts in the root trajectory. These differences suggest that method selection should consider the type of motion and its cues.

The integrated method provided broader detection but did not outperform the top individual method. Its performance would likely benefit from a more refined approach that weighs inputs based on confidence or quality, rather than accepting all candidate points equally.

In summary, although three methods achieved the same similarity value, they did so by identifying different aspects of the motion, offering complementary perspectives. This reinforces the potential value of feature-based segmentation, especially when methods are combined intelligently and tuned to the specific dynamics of the movement being studied.

4.2.4 Comparison with Related Works

Compared to Sonoda et al., whose original framework focused on expressive dance and used linear velocity, direction change, and convex hull volume, our adaptation replaces linear velocity with angular velocity to better capture the biomechanical features of martial arts. This adjustment aligns with the nature of Wushu, which relies heavily on joint-level rotation and short bursts of dynamic activity.

While Sonoda's model is rule-based and interpretable, it assumes rhythmic or expressive transitions, which are less common in martial arts. Our results show that the angular velocity method performs better than the rest of the segmentation methods in this context, highlighting the need to adjust segmentation cues to domain-specific motion characteristics.

Several deep learning-based methods offer strong segmentation performance, such as the MS-GCN model by Filtjens et al., which uses spatial-temporal graphs. However, such models require large, annotated datasets and computational resources. Our rule-based framework is unsupervised, lightweight, and interpretable, making it better suited for small datasets and real-time applications.

Qiu et al. applied angular signal analysis using IMUs for kayaking, showing that angular velocity is a robust segmentation feature in cyclic sports. Our use of angular velocity extends this idea into non-cyclic, fast, high-variation sequences like Wushu. Unlike their sensor-based approach, we operate on optical motion capture data, offering higher spatial resolution without wearable devices.

Other video-based techniques such as those by Wu & Zhou (2024) or Zhang et al. (2022) focus on action classification or localization using CNNs or point-cloud models. These are suited for broader action recognition tasks but are less focused on fine-grained frame-level segmentation. Our method, while simpler, is grounded in biomechanical features and provides interpretable, per-frame segment boundaries, which are critical for structural feedback in martial arts.

Unsupervised filtering-based methods, like the ARMA segmentation by Mei et al. (2021) and Yan et al. (2022), support the effectiveness of rule-driven approaches for motion analysis. These could complement our framework by improving segmentation in lower-energy motion or ambiguous pauses, areas where the current system may still misclassify.

Finally, while synthesis-based models like those by Starke et al. (2021) aim to generate martial arts motion data, they often smooth transitions in a way that loses biomechanical accuracy. Our segmentation approach instead identifies real, meaningful motion boundaries that could improve such models if used to define phase transitions more accurately.

4.2.5 Limitations and Future Works

Despite the promising results from the segmentation methods employed in this study, several key limitations were identified that impacted the accuracy and reliability of the segmentation process. These limitations must be addressed in future work to improve the robustness of the framework.

A primary limitation of the study is the use of a fixed 100-frame tolerance window in evaluating the predicted segmentation points against the ground truth data. This window was based on the assumption that each transitional phase lasts approximately 200 frames. However, this assumption did not hold for all cases. Some transitions occurred in less than 200 frames, with certain transitions lasting only between 50 and 100 frames. As a result, non-transitional phases, such as kicking motions, were sometimes misclassified as transitional phases. Specifically, frames where subtle changes in the movement occurred within the 100-frame tolerance window were incorrectly detected as transitions. While the tolerance window worked

in some cases, it proved to be inadequate for all transitional phases, thus affecting the overall segmentation accuracy.

Another significant limitation emerged from the threshold computation used to detect local minima following local maxima, which was intended to identify transitional phases after a sharp movement, such as the end of a kick. While this approach works well for clear transitions, small "jitters" in the data caused by minor fluctuations or noise sometimes created a sharp drop in the values, leading to false detection of local minima after local maxima. These small jitters, although not reflecting true transitions, were incorrectly identified as valid candidates, resulting in non-transitional frames being marked as part of a transition phase. This issue, especially in Methods 1 (angular velocity) and Method 3 (volume change), resulted in false positives and reduced the segmentation accuracy. These jitter-related errors in threshold detection can lead to frames being falsely labeled as transitional phases, undermining the overall segmentation performance.

Moreover, the segmentation methods were evaluated on a single type of martial arts movement. While this allowed for focused analysis, it limited the generalizability of the findings to other martial arts techniques or even to other movement styles within Wushu. Future studies should aim to expand the dataset to include a wider variety of martial arts techniques, incorporating both different styles and upper-body motions, which are often integral to many martial arts techniques.

Despite these limitations, the study's findings suggest important areas for improvement and refinement. Future work should dynamically adjust the tolerance window to account for variations in transitional phase durations. Additionally, more sophisticated filtering techniques to reduce jitter effects and handle small fluctuations in the data would enhance the reliability of the segmentation. Noise reduction strategies, such as smoothing or signal processing methods, should also be explored to improve the robustness of the threshold computation.

Furthermore, integration methods could be refined by introducing confidence scoring or weighting mechanisms to ensure that the combination of segmentation methods better reflects the strength of each individual technique. This would help prevent the dilution of the results due to weaker methods introducing false positives.

4.2.6 Summary of Findings

This study evaluated whether motion segmentation could be effectively applied to fast, non-repetitive martial arts movements using 3D motion capture data. Specifically, it examined

four segmentation methods—angular velocity, direction change, volume change, and an integrated combination of the three—applied to a structured Wushu kicking sequence.

Using cosine similarity as the evaluation metric, the angular velocity method proved the most reliable, achieving a similarity score of 0.9574 by correctly matching 11 out of 12 annotated transitions. This method effectively captured sharp rotational motion, particularly from leg joints, which often marks clear transitions during Wushu kicks.

The volume change, direction change, and integrated methods each achieved a similarity score of 0.8660, with 9 matched transitions. However, despite having the same numerical similarity, these methods identified different subsets of the ground truth transitions. For example:

- The direction change method failed to detect transitions near frames 880, 2162, and 2265, which typically involved stationary phases or smooth root movement without a sharp turn.
- The volume change method missed transitions at 522, 1595, and 2265, often where posture changes were subtle or spatial configuration stayed consistent despite a phase shift.
- The integrated method, while combining outputs from all three, also missed transitions at 880, 1595, and 2265, showing that simple feature union may not always compensate for individual weaknesses.

These distinctions underscore the unique strengths and limitations of each method, showing that similarity scores alone do not fully capture the performance of segmentation methods. A deeper understanding of which transitions are accurately identified and which are missed provides important context for selecting the most appropriate method.

Overall, the findings suggest that segmentation based on biomechanical features, especially angular velocity, can successfully identify meaningful motion phases in Wushu kicking motions without requiring large labeled datasets or machine learning models. The complementary nature of the different features indicates that integrating methods in a more refined manner, such as using confidence weighting or selective fusion, could further enhance segmentation robustness and improve the detection of transitions across a broader range of martial arts movements.

However, the study also revealed some important limitations. The use of a fixed 100-frame tolerance window, based on the assumption that transitional phases last about 200 frames, was

not universally applicable. Some transitions lasted less than 200 frames, which led to the misclassification of certain non-transitional phases as transitional ones. This limitation, combined with small jitters in the data, particularly in the angular velocity and volume change methods, resulted in false detections of local minima, which were mistakenly interpreted as transitional phases. These issues caused false positives and impacted the overall accuracy of the segmentation.

To address these limitations, future research should focus on dynamically adapting the tolerance window to better align with the actual duration of transitions, as well as developing advanced filtering techniques to mitigate the effects of small fluctuations in the data. Additionally, refining the integration strategies by incorporating more selective weighting or confidence scoring for each method could reduce the noise from weaker signals and improve segmentation accuracy. These refinements would make the framework more reliable and adaptable, not only for Wushu kicking sequences but also for other martial arts and dynamic movement types.

In conclusion, while angular velocity provided the most accurate and reliable segmentation, there is considerable potential for improving the robustness of the segmentation framework by addressing the limitations identified in this study. Future work should aim to refine the tolerance window, reduce jitter-related errors, and develop more effective integration strategies to enhance the overall accuracy and applicability of motion segmentation methods in martial arts and other complex movement analyses.

CHAPTER V
CONCLUSION

5.1 Conclusion

This study explored motion segmentation for martial arts, focusing on Wushu kicking techniques recorded through 3D motion capture. The work adapted a feature-based segmentation framework from Sonoda et al., implementing three interpretable methods of angular velocity, direction change, and volume change, along with an integrated approach that combines all three. The goal was to determine how these methods perform in identifying motion transitions without relying on labeled training data or machine learning.

To address **RQ1** (*How can Sonoda's segmentation methods be adapted for martial arts?*), this research successfully implemented and refined Sonoda's original framework using Wushu motion capture data. Each method was adapted to the movement characteristics of martial arts kicks, particularly high-speed, non-cyclic transitions that are common in Wushu.

In answering **RQ2** (*Which individual method is most accurate?*), the angular velocity method outperformed the others, achieving a dot product similarity of 0.9574 and correctly matching 11 out of 12 annotated transitions. This shows that transitions in Wushu kicks are closely tied to sharp changes in joint rotation. The volume change and direction change methods each scored 0.8660, successfully matching 9 transitions. While they achieved the same similarity value, further analysis showed they captured different ground truth transitions, exposing unique strengths and weaknesses. This underlines that similar scores can mask underlying differences in segmentation behavior.

For **RQ3** (*How does the integrated method compare?*), the combined method also scored 0.8660, matching 9 transitions. It detected some points not captured by any individual method, suggesting that integration can improve coverage. However, it also passed through false positives from weaker methods, limiting its performance. Without proper weighting or filtering, simple union-based fusion does not guarantee improvement over the best individual method.

To address **RQ4** (*How do the segmentation results compare to expert-annotated ground truth?*), a greedy matching process with a 100-frame tolerance was used, and the cosine similarity metric provided a clear, interpretable way to compare predictions with ground truth. Angular velocity aligned most strongly with expert-annotated transitions, while other methods showed more variability and selective effectiveness depending on movement type.

Despite the promising results, several limitations impacted the segmentation process. One of the primary limitations was the use of a fixed 100-frame tolerance window in evaluating segmentation results. This tolerance window, based on the assumption that transitional phases typically last around 200 frames, was not universally applicable. Some transitions were shorter in duration, leading to misclassification of non-transitional phases, such as kicks, as transitional. This caused some errors in the final segmentation. Additionally, small jitters in the data—particularly in the angular velocity and volume change methods—led to erroneous local minima detections, which were incorrectly interpreted as transitional phases. These minor fluctuations in the data, though not reflecting true transitions, were mistakenly marked as part of the transition phase, introducing false positives and reducing segmentation accuracy. Finally, the focus on a single Wushu kicking sequence limited the scope and generalizability of the findings. Future research should aim to expand the dataset to include a wider variety of martial arts techniques and include upper-body movements, as these are integral to many martial arts techniques and transitions.

In conclusion, this study shows that biomechanically grounded segmentation methods, particularly those based on rotational motion (angular velocity), can effectively detect meaningful motion phases in Wushu kicking motions without the need for large labeled datasets or complex learning-based models. The complementary nature of different features suggests that refining integration strategies—such as confidence scoring or selective fusion—can improve segmentation robustness and accuracy across a broader range of movements. The findings contribute to the development of a lightweight, interpretable framework for motion analysis, which can be useful for performance feedback systems and further research in martial arts and biomechanics.

5.2 Suggestions

Based on the findings and limitations of this study, several suggestions can be proposed to guide future research and practical implementation:

1. Refine the Evaluation Tolerance Window Approach

The use of a fixed 100-frame tolerance window caused misclassifications for transitions shorter than expected. Future work should dynamically adjust the tolerance window to account for variations in transitional phase durations, ensuring more accurate segmentation for different movements.

2. Improve Threshold Computation for Jitter Mitigation

Threshold computation occasionally led to false positives due to small jitters in the data. Future research should implement smoothing or filtering techniques to reduce jitter effects and adaptive thresholding to better distinguish meaningful transitions from noise.

3. Expand Dataset Diversity

The current analysis is based on a single Wushu kicking sequence. Future work should explore a broader dataset that includes various techniques (e.g., punches, stances, transitions), multiple performers, and different martial arts styles such as Karate, Silat, or Taekwondo. This would improve the generalizability of the segmentation framework.

4. Incorporate Upper Body Motion

While this research focused on lower-limb dynamics, many martial arts techniques involve complex coordination of upper-body movements. Including shoulder, arm, and torso markers in the segmentation process could offer a more comprehensive understanding of full-body transitions.

5. Refine the Integration Mechanism

The current integrated method uses a simple logical combination of features. Future improvements could involve weighted fusion, confidence scoring, or adaptive filtering to reduce false positives and better leverage the strengths of each individual method.

6. Compare with Learning-Based Models

While this study focused on interpretable, rule-based segmentation, it would be valuable to benchmark its performance against modern learning-based models (e.g., MS-GCN, DT-FCN) on the same dataset. Hybrid approaches may also be explored, where rule-based features serve as inputs or priors in learning pipelines.

LITERATURE

Carnegie Mellon University. (n.d.). *Marker Placement Guide*. Retrieved July 20, 2025, from https://mocap.cs.cmu.edu/markerPlacementGuide.pdf

Cheema, N., Hosseini, S., Sprenger, J., Herrmann, E., Du, H., Fischer, K., & Slusallek, P. (2019). *Fine-Grained Semantic Segmentation of Motion Capture Data using Dilated Temporal Fully-Convolutional Networks*. http://arxiv.org/abs/1903.00695

Ciaccioni, S., Castro, O., Bahrami, F., Tomporowski, P. D., Capranica, L., Biddle, S. J. H., Vergeer, I., & Pesce, C. (2024). Martial arts, combat sports, and mental health in adults: A systematic review. In *Psychology of Sport and Exercise* (Vol. 70). Elsevier Ltd. https://doi.org/10.1016/j.psychsport.2023.102556

Echeverria, J., & Santos, O. C. (2021). Toward modeling psychomotor performance in karate combats using computer vision pose estimation. *Sensors*, *21*(24). https://doi.org/10.3390/s21248378

Filtjens, B., Vanrumste, B., & Slaets, P. (2022). *Skeleton-Based Action Segmentation with Multi-Stage Spatial-Temporal Graph Convolutional Neural Networks*. https://github.com/BenjaminFiltjens/MS-GCN.

Fod, A., Mataric, M. J., & Jenkins, O. C. (2002). Automated Derivation of Primitives for Movement Classification. In *Autonomous Robots* (Vol. 12, Issue 1). http://robotics.usc.edu/˜agents/imitation.html

Han, Q. S., Theeboom, M., & Zhu, D. (2021). Chinese martial arts and the Olympics: Analysing the policy of the International Wushu Federation. *International Review for the Sociology of Sport*, *56*(5), 603–624. https://doi.org/10.1177/1012690220957177

Hu, X., Bao, X., Xie, S., & Wei, G. (2021). Unsupervised motion capture data segmentation based on topic model. *Computer Animation and Virtual Worlds*, *32*(3–4). https://doi.org/10.1002/cav.2005

Mei, F., Hu, Q., Yang, C., & Liu, L. (2021). Arma-based segmentation of human limb motion sequences. *Sensors*, *21*(16). https://doi.org/10.3390/s21165577

Menolotto, M., Komaris, D. S., Tedesco, S., O'flynn, B., & Walsh, M. (2020). Motion capture technology in industrial applications: A systematic review. In *Sensors (Switzerland)* (Vol. 20, Issue 19, pp. 1–25). MDPI AG. https://doi.org/10.3390/s20195687

Qiu, S., Hao, Z., Wang, Z., Liu, L., Liu, J., Zhao, H., & Fortino, G. (2022). Sensor Combination Selection Strategy for Kayak Cycle Phase Segmentation Based on Body Sensor Networks.

IEEE Internet of Things Journal, *9*(6), 4190–4201. https://doi.org/10.1109/JIOT.2021.3102856

Schiff, N. T. (2022). *Ethical Values in The Martial Art of Pencak Silat*.

SNS Insider. (2023). *3D Motion Capture System Market Size, Share & Segmentation By Systems (Optical Systems and Non-optical Systems), By Component (Software, Hardware, and Services), By Application (Engineering & Design, Media & Entertainment, Biomedical Research, Industrial, Education, and Others), By Regions And Global Forecast 2024-2032*. https://www.snsinsider.com/reports/3d-motion-capture-system-market-3416

Sonoda, M., Tsuruta, S., & Yoshimura, M. (2008). *Segmentation of Dancing Movement by Extracting Features from Motion Capture Data*.

Stamenković, A., Manić, M., Roklicer, R., Trivić, T., Malović, P., & Drid, P. (2022). Effects of Participating in Martial Arts in Children: A Systematic Review. In *Children* (Vol. 9, Issue 8). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/children9081203

Starke, S., Zhao, Y., Zinno, F., & Komura, T. (2021). Neural animation layering for synthesizing martial arts movements. *ACM Transactions on Graphics*, *40*(4), 1–16. https://doi.org/10.1145/3450626.3459881

Szczęsna, A., Błaszczyszyn, M., & Pawlyta, M. (2021). Optical motion capture dataset of selected techniques in beginner and advanced Kyokushin karate athletes. *Scientific Data*, *8*(1). https://doi.org/10.1038/s41597-021-00801-5

Wan Idris, W. M. R., Rafi, A., Bidin, A., Jamal, A. A., & Fadzli, S. A. (2019). A systematic survey of martial art using motion capture technologies: the importance of extrinsic feedback. *Multimedia Tools and Applications*, *78*(8), 10113–10140. https://doi.org/10.1007/s11042-018-6624-y

Wang, X., Yu, H., Kold, S., Rahbek, O., & Bai, S. (2023). Wearable sensors for activity monitoring and motion control: A review. In *Biomimetic Intelligence and Robotics* (Vol. 3, Issue 1). Elsevier B.V. https://doi.org/10.1016/j.birob.2023.100089

Wu, B., & Zhou, J. (2024). Video-Based Martial Arts Combat Action Recognition and Position Detection Using Deep Learning. *IEEE Access*, *12*, 161357–161374. https://doi.org/10.1109/ACCESS.2024.3487289

Yan, S., Chen, J., & Huang, H. (2022). Biomechanical Analysis of Martial Arts Movements Based on Improved PSO Optimized Neural Network. *Mobile Information Systems*, *2022*. https://doi.org/10.1155/2022/8189426

Yangyang, D. (2023). A Study of the Social Function of Chinese Wushu. *Academic Journal of Humanities & Social Sciences*, *6*(12). https://doi.org/10.25236/ajhss.2023.061211

Zhang, X., Wu, X., & Song, L. (2022). Arm Movement Analysis Technology of Wushu Competition Image Based on Deep Learning. *Computational Intelligence and Neuroscience*, *2022*. https://doi.org/10.1155/2022/9866754

Zhou, H. (2022). Martial Arts Moves Recognition Method Based on Visual Image. *Journal of Information Processing Systems*, *18*(6), 813–821. https://doi.org/10.3745/JIPS.02.0188

APPENDIX

Appendix A. MATLAB Code Snippet for Preprocessing Wushu Data

wushu_skeleton_FINAL.m

```
orang = readcell('Wushu data coordinates - modified.xlsx');
headers = orang(1, :);
combined_cols = 2:4:size(orang, 2);
body_parts = headers(combined_cols);

num_parts = length(combined_cols);
num_frames = size(orang, 1) - 1;

x_coords_all = nan(num_frames, num_parts);
y_coords_all = nan(num_frames, num_parts);
z_coords_all = nan(num_frames, num_parts);

for frame_row = 2:num_frames+1
    for i = 1:num_parts
        col = combined_cols(i);
        data_col = orang(frame_row, col);
        data_str = cell2mat(data_col);

        data_str = strrep(data_str, '(', '');
        data_str = strrep(data_str, ')', '');
        coords = strsplit(data_str, ',');

        if length(coords) == 3
            y_coords_all(frame_row-1, i) = str2double(coords{1});
            z_coords_all(frame_row-1, i) = str2double(coords{2});
            x_coords_all(frame_row-1, i) = str2double(coords{3});
        end
    end
end

for i = 1:num_parts
    valid_idx = find(x_coords_all(:, i) ~= 0 & y_coords_all(:, i) ~= 0 & z_coords_all(:, i) ~= 0);

    if ~isempty(valid_idx)
        x_coords_all(:, i) = interp1(valid_idx, x_coords_all(valid_idx, i), 1:num_frames, 'linear', 'extrap');
        y_coords_all(:, i) = interp1(valid_idx, y_coords_all(valid_idx, i), 1:num_frames, 'linear', 'extrap');
        z_coords_all(:, i) = interp1(valid_idx, z_coords_all(valid_idx, i), 1:num_frames, 'linear', 'extrap');
    end
end
```

Appendix B. MATLAB Code Snippet for Visualizing Wushu Data

wushu_skeleton_FINAL.m *

```
figure('Position', [500, 100, 1200, 900]);
hold on;
grid on;
axis equal;
xlabel('X-axis'); ylabel('Y-axis'); zlabel('Z-axis');
xlim([-1500, 1500]);
ylim([-4000, 1000]);
zlim([0, 2000]);
view(45, 10);
rotate3d on;

scatter_plot = scatter3([], [], [], 25, 'filled');
line_plots = gobjects(length(connections), 1);

for i = 1:length(connections)
    line_plots(i) = plot3([0, 0], [0, 0], [0, 0], 'k-', 'LineWidth', 2);
end

% Animation loop (max 2378)

for frame_idx = 2265
    x_coords = x_coords_all(frame_idx, :);
    y_coords = y_coords_all(frame_idx, :);
    z_coords = z_coords_all(frame_idx, :);

    set(scatter_plot, 'XData', x_coords, 'YData', y_coords, 'ZData', z_coords);

    for i = 1:length(connections)
        marker1 = connections{i}{1};
        marker2 = connections{i}{2};
        idx1 = find(strcmp(body_parts, marker1));
        idx2 = find(strcmp(body_parts, marker2));

        if ~isempty(idx1) && ~isempty(idx2)
            set(line_plots(i), 'XData', [x_coords(idx1), x_coords(idx2)], ...
                               'YData', [y_coords(idx1), y_coords(idx2)], ...
                               'ZData', [z_coords(idx1), z_coords(idx2)]);
        end
    end

    title(['3D Skeleton Visualization - Frame ', num2str(frame_idx)]);
    pause(0.01);
end

saveas(gcf, 'Wushu_GroundTruth_Frame.png');
hold off;
```

Appendix C. MATLAB Code Snippet for Segmentation Based on Angular Velocity

wushu_AngularVelocity_FINAL.m

```
marker_list = {'RBWT', 'RFWT', 'RHIP', 'RANK', 'RTOE', 'LBWT', 'LFWT', 'LHIP', 'LANK', 'LTOE'};
marker_idx = containers.Map();

for i = 1:numel(marker_list)
    idx = find(strcmp(body_parts, marker_list{i}));
    if isempty(idx)
        error('Marker %s not found', marker_list{i});
    else
        marker_idx(marker_list{i}) = idx;
    end
end

angle_between = @(u, v) acos(max(min(dot(u, v) / (norm(u) * norm(v)), 1), -1));
theta_all = nan(num_frames, 6);

for frame_idx = 1:num_frames
    RBWT = [x_coords_all(frame_idx, marker_idx('RBWT')), y_coords_all(frame_idx, marker_idx('RBWT')), z_coords_all(frame_idx, marker_idx('RBWT'))];
    RFWT = [x_coords_all(frame_idx, marker_idx('RFWT')), y_coords_all(frame_idx, marker_idx('RFWT')), z_coords_all(frame_idx, marker_idx('RFWT'))];
    RHIP = [x_coords_all(frame_idx, marker_idx('RHIP')), y_coords_all(frame_idx, marker_idx('RHIP')), z_coords_all(frame_idx, marker_idx('RHIP'))];
    RANK = [x_coords_all(frame_idx, marker_idx('RANK')), y_coords_all(frame_idx, marker_idx('RANK')), z_coords_all(frame_idx, marker_idx('RANK'))];
    RTOE = [x_coords_all(frame_idx, marker_idx('RTOE')), y_coords_all(frame_idx, marker_idx('RTOE')), z_coords_all(frame_idx, marker_idx('RTOE'))];
    LBWT = [x_coords_all(frame_idx, marker_idx('LBWT')), y_coords_all(frame_idx, marker_idx('LBWT')), z_coords_all(frame_idx, marker_idx('LBWT'))];
    LFWT = [x_coords_all(frame_idx, marker_idx('LFWT')), y_coords_all(frame_idx, marker_idx('LFWT')), z_coords_all(frame_idx, marker_idx('LFWT'))];
    LHIP = [x_coords_all(frame_idx, marker_idx('LHIP')), y_coords_all(frame_idx, marker_idx('LHIP')), z_coords_all(frame_idx, marker_idx('LHIP'))];
    LANK = [x_coords_all(frame_idx, marker_idx('LANK')), y_coords_all(frame_idx, marker_idx('LANK')), z_coords_all(frame_idx, marker_idx('LANK'))];
    LTOE = [x_coords_all(frame_idx, marker_idx('LTOE')), y_coords_all(frame_idx, marker_idx('LTOE')), z_coords_all(frame_idx, marker_idx('LTOE'))];

    u1 = RHIP - RBWT; v1 = RANK - RHIP;
    u2 = RHIP - RFWT; v2 = RANK - RHIP;
    u3 = RANK - RHIP; v3 = RTOE - RANK;
    u4 = LHIP - LBWT; v4 = LANK - LHIP;
    u5 = LHIP - LFWT; v5 = LANK - LHIP;
    u6 = LANK - LHIP; v6 = LTOE - LANK;

    theta_all(frame_idx, 1) = angle_between(u1, v1);
    theta_all(frame_idx, 2) = angle_between(u2, v2);
    theta_all(frame_idx, 3) = angle_between(u3, v3);
    theta_all(frame_idx, 4) = angle_between(u4, v4);
    theta_all(frame_idx, 5) = angle_between(u5, v5);
    theta_all(frame_idx, 6) = angle_between(u6, v6);
end

dt = 0.01; % 100 FPS
theta_dot_all = nan(num_frames, 6);
```

```
for frame_idx = 2:num_frames
    for angle_idx = 1:6
        if ~isnan(theta_all(frame_idx, angle_idx)) && ~isnan(theta_all(frame_idx - 1, angle_idx))
            theta_dot_all(frame_idx, angle_idx) = ...
                (theta_all(frame_idx, angle_idx) - theta_all(frame_idx - 1, angle_idx)) / dt;
        end
    end
end

z = sum(theta_dot_all.^2, 2);

z_range = max(z, [], 'omitnan') - min(z, [], 'omitnan');
tau = max(0.1 * z_range, 100);

[~, peak_locs] = findpeaks(z, 'MinPeakProminence', tau, 'MinPeakDistance', 10);

[~, min_locs] = findpeaks(-z);

seg_points_AngVel = [];
for i = 1:length(peak_locs)
    next_min = min_locs(find(min_locs > peak_locs(i), 1));
    if ~isempty(next_min) && z(next_min) < tau
        seg_points_AngVel(end+1) = next_min;
    end
end

figure;
plot(z); hold on;
stem(seg_points_AngVel, z(seg_points_AngVel), 'r', 'filled', 'Marker', 'o');
xlabel('Frame');
ylabel('Sum of Squared Angular Velocity (rad/s)^2');
title('Segmentation by Angular Velocity');
grid on;

fprintf('Method 1 - Detected Candidate Frames (Angular Velocity):\n');
disp(seg_points_AngVel);
fprintf('Total Candidates: %d\n', numel(seg_points_AngVel));

saveas(gcf, 'Wushu_AngVel.png');
```

Appendix D. MATLAB Code Snippet for Segmentation Based on Direction Change

```
wushu_DirectionChange_FINAL.m
/MATLAB Drive/wushu_DirectionChange_FINAL.m
root_marker_name = 'LWRT';
root_idx = find(strcmp(body_parts, root_marker_name));

if isempty(root_idx)
    error('Root marker "LWRT" not found in the body_parts.');
end

L = 15;             % Frame offset for vector computation
T = 30;             % Threshold in mm
num_frames_valid = num_frames - 2*L;
curvature = nan(num_frames, 1);

for i = (L+1):(num_frames-L)
    % Vectors in XY plane
    a = [x_coords_all(i, root_idx) - x_coords_all(i-L, root_idx), ...
         y_coords_all(i, root_idx) - y_coords_all(i-L, root_idx)];

    b = [x_coords_all(i+L, root_idx) - x_coords_all(i, root_idx), ...
         y_coords_all(i+L, root_idx) - y_coords_all(i, root_idx)];

    if norm(a) < T || norm(b) < T
        continue;
    end

    curvature(i) = dot(a, b) / (norm(a) * norm(b));
end

curv_threshold = 0;
seg_points_DirCha = [];
for i = (L+2):(num_frames-L-1)
    if ~isnan(curvature(i)) && ...
   curvature(i-1) > curvature(i) && ...
   curvature(i) < curvature(i+1) && ...
   curvature(i) < curv_threshold
        seg_points_DirCha(end+1) = i;
    end
end

valid_idx = ~isnan(curvature);
```

```
figure;
plot(find(valid_idx), curvature(valid_idx)); hold on;
stem(seg_points_DirCha, curvature(seg_points_DirCha), 'r', 'filled', 'Marker', 'o');
xlabel('Frame');
ylabel('Digital Curvature');
title('Segmentation by Change of Movement Direction');

fprintf('Method 2 - Detected Candidate Frames (Direction Change):\n');
disp(seg_points_DirCha);
fprintf('Total Candidates: %d\n', numel(seg_points_DirCha));

saveas(gcf, 'Wushu_DirCha.png');
```

Appendix E. MATLAB Code Snippet for Segmentation Based on Volume Change

```
wushu_VolumeChange_FINAL.m
/MATLAB Drive/wushu_VolumeChange_FINAL.m
convex_volumes = nan(num_frames, 1);

for frame_idx = 1:num_frames
    coords = [x_coords_all(frame_idx, :)', y_coords_all(frame_idx, :)', z_coords_all(frame_idx, :)'];
    coords = coords(~any(isnan(coords), 2), :);

    if size(coords, 1) >= 4
        try
            [~, volume] = convhull(coords);
            convex_volumes(frame_idx) = volume;
        catch
            convex_volumes(frame_idx) = NaN;
        end
    end
end

volume_diff = abs(diff(convex_volumes));
volume_diff(end + 1) = volume_diff(end);

t = max(0.23 * range(volume_diff, 'omitnan'), 2*10^6);
seg_points_VolCha = [];
for i = 2:num_frames-1
    if volume_diff(i-1) > volume_diff(i) && volume_diff(i) < volume_diff(i+1)
        if volume_diff(i) < t && volume_diff(i+1) > t
            seg_points_VolCha(end + 1) = i;
        end
    end
end

figure;
plot(volume_diff);
hold on;
stem(seg_points_VolCha, volume_diff(seg_points_VolCha), 'r', 'filled', 'Marker', 'o');
xlabel('Frame');
ylabel('Change of Volume');
title('Segmentation by Change of Volume');
grid on;

fprintf('Pure Sonoda Method - Candidate Frames (Volume Change):\n');
disp(seg_points_VolCha);
fprintf('Total Candidates: %d\n', numel(seg_points_VolCha));

saveas(gcf, 'Wushu_VolCha.png');
```

Appendix F. MATLAB Code Snippet for Integration of 3 Methods

wushu_SegmentedFrames.m

```
%% Combine the three methods
all_candidates = unique([seg_points_AngVel, seg_points_DirCha, seg_points_VolCha]);

all_candidates = sort(all_candidates);
combined_seg_points = [];
min_combined_distance = 100;  % minimum distance between points

last_added = -Inf;
for i = 1:length(all_candidates)
    if all_candidates(i) - last_added >= min_combined_distance
        combined_seg_points(end+1) = all_candidates(i);
        last_added = all_candidates(i);
    end
end

method_labels = {'Integrated Segmentation', ...
                 'Method 3: Volume Change', ...
                 'Method 2: Direction Change', ...
                 'Method 1: Angular Velocity'};

raster_data = {combined_seg_points, seg_points_VolCha, seg_points_DirCha, seg_points_AngVel};

figure;
hold on;

for i = 1:length(raster_data)
    y_val = i;
    frames = raster_data{i};
    for j = 1:length(frames)
        x = frames(j);
        plot([x x], [y_val-0.4 y_val+0.4], 'k', 'LineWidth', 1);
    end
end
```

```
% Formatting
yticks(1:length(method_labels));
yticklabels(method_labels);
xlabel('Frame Number');
title('Raster Plot of Segmentation Frames');
ylim([0.5, length(method_labels)+0.5]);
grid on;

fprintf('Method 1 - Detected Candidate Frames (Angular Velocity):\n');
disp(seg_points_AngVel);
fprintf('Total Candidates: %d\n', numel(seg_points_AngVel));

fprintf('Method 2 - Detected Candidate Frames (Direction Change):\n');
disp(seg_points_DirCha);
fprintf('Total Candidates: %d\n', numel(seg_points_DirCha));

fprintf('Method 3 - Detected Candidate Frames (Volume Change):\n');
disp(seg_points_VolCha);
fprintf('Total Candidates: %d\n', numel(seg_points_VolCha));

fprintf('Combined Segmentation Points:\n');
disp(combined_seg_points);
fprintf('Total Combined Points: %d\n', numel(combined_seg_points));

saveas(gcf, 'Wushu_RasterPlot.png');

% Prepare column headers
headers = {'Methods', 'Angular Velocity', 'Direction Change', 'Volume Change', 'Integrated'};

% Determine max number of rows needed
max_len = max([length(seg_points_AngVel), ...
               length(seg_points_DirCha), ...
               length(seg_points_VolCha), ...
               length(combined_seg_points)]);

% Preallocate cell array
export_data = cell(max_len + 1, 5);  % +1 for header
export_data(1, :) = headers;         % Header row

% Fill in segmentation frames under respective columns
for i = 1:max_len
    if i <= length(seg_points_AngVel)
        export_data{i+1, 2} = seg_points_AngVel(i);
    end
    if i <= length(seg_points_DirCha)
        export_data{i+1, 3} = seg_points_DirCha(i);
    end
    if i <= length(seg_points_VolCha)
        export_data{i+1, 4} = seg_points_VolCha(i);
    end
    if i <= length(combined_seg_points)
        export_data{i+1, 5} = combined_seg_points(i);
    end
end

% Export to Excel
output_filename = 'wushu_segmentationResults.xlsx';
writetable(cell2table(export_data), output_filename, 'WriteVariableNames', false);

disp(['Segmentation results exported to ', output_filename]);
```

Appendix G. MATLAB Code Snippet for Computing Ground Truth

wushu_groundTruth_accuracyTest_FINAL.m

```
clc;
clear;

% --- Load and Process Ground Truth ---
filename = 'wushu_groundTruth.xlsx';
raw_gt = readmatrix(filename);

all_gt_frames = raw_gt(:);
all_gt_frames = all_gt_frames(~isnan(all_gt_frames));
all_gt_frames = sort(all_gt_frames);

fprintf('--- Raw Ground Truth Frames ---\n');
disp(all_gt_frames');
fprintf('Total Ground Truth Points (raw, unfiltered): %d\n\n', numel(all_gt_frames));

% --- Filter Ground Truth using a tolerance ---
frame_tolerance = 100;
filtered_gt = [];

for i = 1:length(all_gt_frames)
    current_frame = all_gt_frames(i);
    matches = abs(all_gt_frames - current_frame) <= frame_tolerance;
    if sum(matches) >= 2
        filtered_gt(end+1) = current_frame;
    end
end

filtered_gt = sort(unique(filtered_gt));

% --- Ensure minimum separation between ground truth points ---
final_gt = [];
last_added = -Inf;
min_separation = 100;
```

```
for i = 1:length(filtered_gt)
    if filtered_gt(i) - last_added > min_separation
        final_gt(end+1) = filtered_gt(i);
        last_added = filtered_gt(i);
    end
end

% --- Define Method Names and Predicted Segmentation Points ---
method_names = {
    'Angular Velocity', ...
    'Direction Change', ...
    'Volume Change', ...
    'Integrated'
};

% These variables (seg_points_AngVel, seg_points_DirCha, seg_points_VolCha, combined_seg_points)
% should already be defined in your workspace. For example:
seg_points_AngVel = [78, 276, 434, 620, 814, 984, 1140, 1338, 1502, 1885, 1953, 2118];
seg_points_DirCha = [252, 430, 602, 775, 981, 1134, 1344, 1409, 1515, 1827, 2006];
seg_points_VolCha = [160, 251, 416, 638, 775, 831, 1038, 1317, 1416, 1914, 2112];
combined_seg_points = [78, 251, 416, 602, 775, 981, 1134, 1317, 1502, 1827, 1953, 2112];

predicted_methods = {
    seg_points_AngVel, ...
    seg_points_DirCha, ...
    seg_points_VolCha, ...
    combined_seg_points
};

fprintf('\nFiltered Ground Truth (final_gt):\n');
disp(final_gt);
fprintf('Total Ground Truth Points: %d\n', numel(final_gt));
```

```
% --- Dot Product Similarity Evaluation ---

fprintf('\n--- Dot Product Similarity Evaluation ---\n');

dot_similarities = zeros(1, length(method_names));

for m = 1:length(method_names)
    predicted = predicted_methods{m};
    match_count = 0;
    matched_gt = false(size(final_gt));

    for i = 1:length(predicted)
        for j = 1:length(final_gt)
            if ~matched_gt(j) && abs(predicted(i) - final_gt(j)) <= frame_tolerance
                match_count = match_count + 1;
                matched_gt(j) = true;
                break;
            end
        end
    end

    % Dot product similarity formula
    if match_count == 0
        similarity = 0;
    else
        similarity = match_count / (sqrt(match_count) * sqrt(length(final_gt)));
    end

    dot_similarities(m) = similarity;
```

```
    fprintf('\nMethod: %s\n', method_names{m});
    fprintf('Total Predictions: %d\n', length(predicted));
    fprintf('Matched Predictions (dot product): %d\n', match_count);
    fprintf('Dot Product Similarity: %.4f\n', similarity);
end

% --- Plot Dot Product Similarity Scores ---
figure;
bar_colors = [
    0.2 0.6 0.8;  % Angular Velocity - blue
    0.9 0.4 0.3;  % Direction Change - red-orange
    0.3 0.7 0.4;  % Volume Change - green
    0.6 0.4 0.8   % Integrated - purple
];

b = bar(dot_similarities, 'FaceColor', 'flat');
ylim([0 1]);
title('Dot Product Similarity');
ylabel('Similarity Score');
set(gca, 'XTickLabel', method_names, 'FontSize', 10);
grid on;

for i = 1:length(method_names)
    b.CData(i, :) = bar_colors(i, :);
end

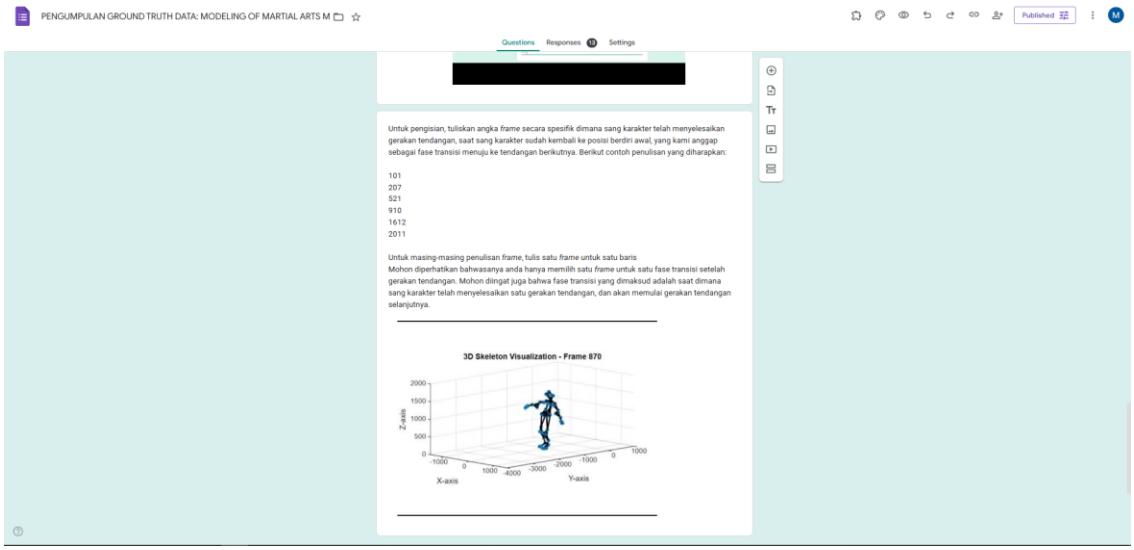
saveas(gcf, 'Wushu_groundTruth_dotProductSimilarity.png');
```

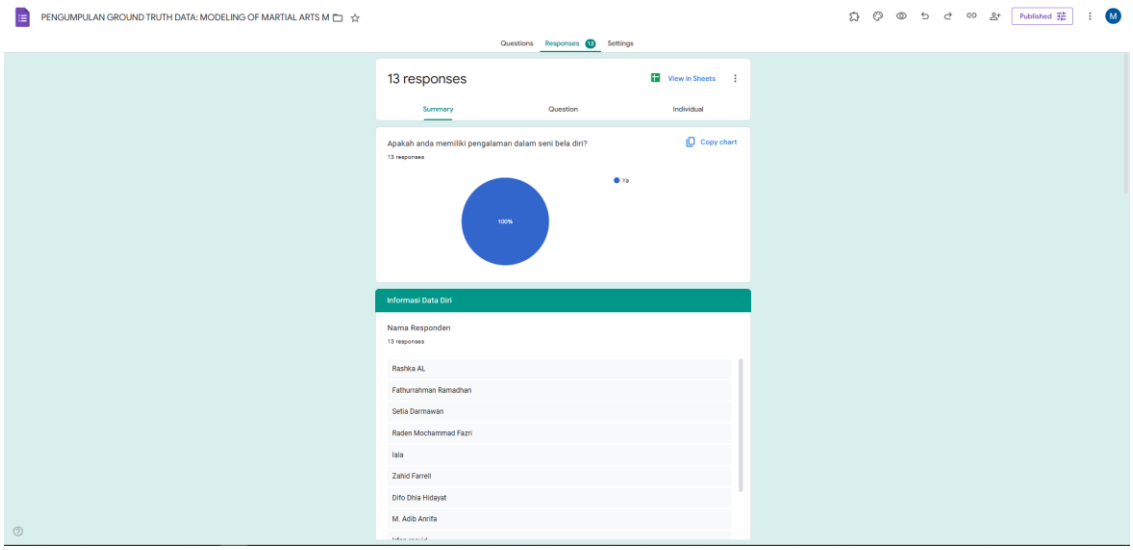
Appendix H. Dataset Markers and their Anatomical Location

Marker Name	**Anatomical Location**	**Marker Name**	**Anatomical Location**
ARIEL	Top of head (center)	LFHD	Left front of head
LBHD	Left back of head	RFHD	Right front of head
RBHD	Right back of head	C7	Base of neck (C7 vertebra)

T10	Mid back (T10 vertebra)	CLAV	Clavicle
STRN	Sternum	LFSH	Left front shoulder
LBSH	Left back shoulder	LUPA	Left upper arm
LELB	Left elbow	LIEL	Left inner elbow
LOWR	Left outer wrist	LIWR	Left inner wrist
LWRE	Left wrist end	LIHAND	Left index base (hand)
LOHAND	Left outer hand	RFSH	Right front shoulder
RBSH	Right back shoulder	RUPA	Right upper arm
RELB	Right elbow	RIEL	Right inner elbow
ROWR	Right outer wrist	RIWR	Right inner wrist
RWRE	Right wrist end	RIHAND	Right index base (hand)
ROHAND	Right outer hand	LFWT	Left front waist
LMWT	Left mid waist	LBWT	Left back waist
RFWT	Right front waist	RMWT	Right mid waist
RBWT	Right back waist	LHIP	Left hip
LKNE	Left knee	LKNI	Left knee inner
LSHN	Left shin	LANK	Left ankle
LHEL	Left heel	LMT5	Left metatarsal 5 (outer foot)
LMT1	Left metatarsal 1 (inner foot)	LTOE	Left toe
RHIP	Right hip	RKNE	Right knee
RKNI	Right knee inner	RSHN	Right shin
RANK	Right ankle	RHEL	Right heel
RMT5	Right metatarsal 5 (outer foot)	RMT1	Right metatarsal 1 (inner foot)
RTOE	Right toe		

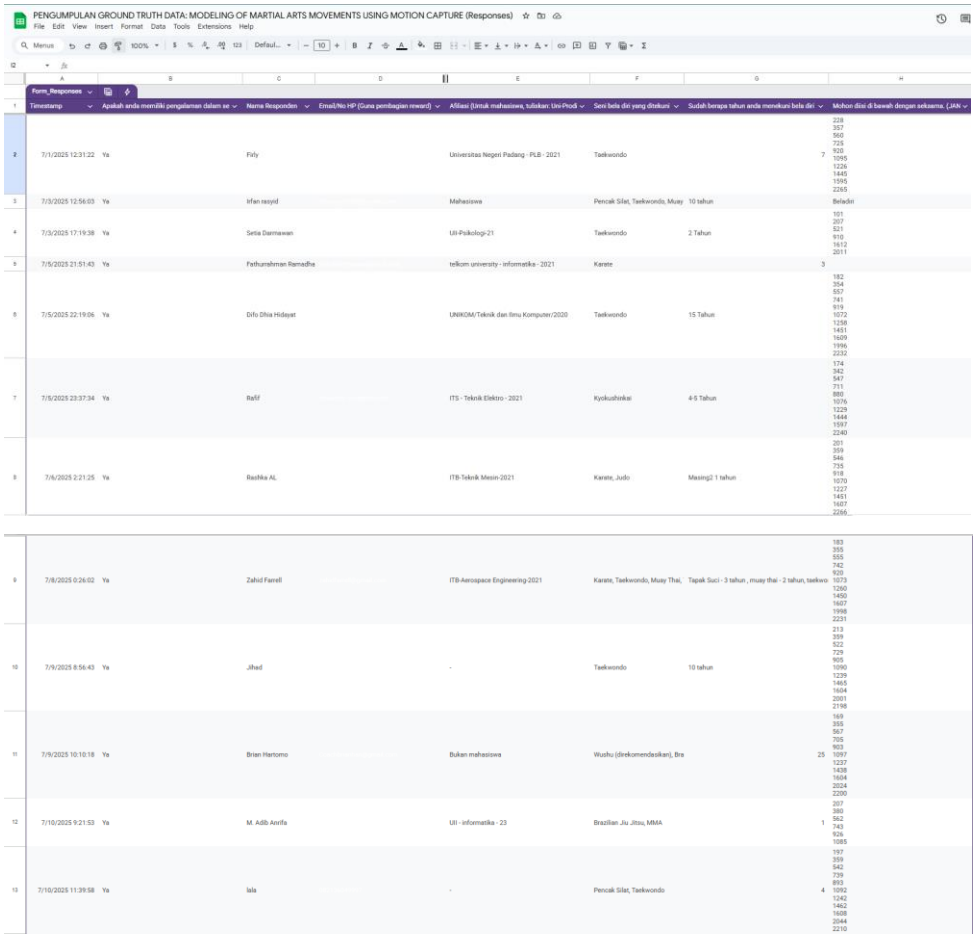
Appendix I. Google Form for Collecting Ground Truth Data







Appendix J. Collected Responds from Google Form



PENGUMPULAN GROUND TRUTH DATA: MODELING OF MARTIAL ARTS MOVEMENTS USING MOTION CAPTURE (Responses)

File Edit View Insert Format Data Tools Extensions Help

Form_Responses

	Timestamp	Apakah anda memiliki pengalaman dalam se	Nama Responden	Email/No HP (Guna pembagian reward)	Afiliasi (Untuk mahasiswa, tuliskan: Uni-Prodi	Seni bela diri yang ditekuni	Sudah berapa tahun anda menekuni bela diri	Mohon diisi di bawah dengan sekuens. (JAN
2	7/1/2025 12:31:22	Ya	Firly		Universitas Negeri Padang - PLB - 2021	Taekwondo	7	228 357 560 725 920 1095 1226 1445 1595 2265
3	7/3/2025 12:56:03	Ya	Irfan rasyid		Mahasiswa	Pencak Silat, Taekwondo, Muay	10 tahun	Beladiri
4	7/3/2025 17:19:38	Ya	Setia Darmawan		UII-Psikologi-21	Taekwondo	2 Tahun	101 207 521 910 1612 2011
9	7/5/2025 21:51:43	Ya	Fathurrahman Ramadha		telkom university - informatika - 2021	Karate	3	
6	7/5/2025 22:19:06	Ya	Difo Dhia Hidayat		UNIKOM/Teknik dan Ilmu Komputer/2020	Taekwondo	1S Tahun	182 354 557 741 919 1072 1258 1451 1609 1996 2232
7	7/5/2025 23:37:34	Ya	Rafif		ITS - Teknik Elektro - 2021	Kyokushinkai	4-5 Tahun	174 342 547 711 880 1076 1229 1444 1597 2240
8	7/6/2025 2:21:25	Ya	Rashka AL		ITB-Teknik Mesin-2021	Karate, Judo	Masing2 1 tahun	201 359 546 715 918 1070 1227 1451 1607 2266
9	7/8/2025 0:26:02	Ya	Zahid Farrell		ITB-Aerospace Engineering-2021	Karate, Taekwondo, Muay Thai,	Tapak Suci - 3 tahun , muay thai - 2 tahun, taekwo	183 355 555 742 920 1073 1260 1450 1607 1998 2231
10	7/9/2025 8:56:43	Ya	Jihad		-	Taekwondo	10 tahun	213 359 522 729 905 1090 1239 1465 1604 2001 2198
11	7/9/2025 10:10:18	Ya	Brian Hartomo		Bukan mahasiswa	Wushu (direkomendasikan), Bra	25	169 355 567 765 903 1097 1237 1438 1604 2024 2200
12	7/10/2025 9:21:53	Ya	M. Adib Anrifa		UII - informatika - 23	Brazilian Jiu Jitsu, MMA	1	207 380 562 743 926 1085
13	7/10/2025 11:39:58	Ya	lala		-	Pencak Silat, Taekwondo	4	197 359 542 739 893 1092 1242 1462 1608 2044 2210