Development of Fall Detection System Using Depth Camera for Monitoring Elderly People Living Alone

- Inverted pendulum model to discriminate falling motion -

Eisuke TOMITA[1] and Akinori SEKIGUCHI[2]

[1] Sustainable Engineering Program, Tokyo University of Techonology Graduate School, Hachiouji, Tokyo, 192-0982, Japan
e-mail:g5123053e0@edu.teu.ac.jp
[2] Department of Mechanical Engineering, Tokyo University of Techonology, Hachiouji, Tokyo, 192-0982, Japan
e-mail:sekiguchi@stf.teu.ac.jp

Abstract. The purpose of this study is to develop a fall detection system for monitoring elderly individuals living alone, and to use simulation to create training data for the system. We developed a system using a depth camera to calculate the body's acceleration. By performing actual falling motions, we confirmed that sudden accelerations occur during falls. We implemented a triple inverted pendulum, commonly used for detailed analyzing human motion, in a dynamics simulator and conducted simulations of falling and non-falling movements. As a result, we confirmed that the obtained time-series data we generated closely resembled the actual measured motions. Using the time-series data obtained from the triple inverted pendulum simulation as training data, we applied machine learning to classify falling and non-falling motions measured with Azure Kinect. As a result, we confirmed that each type of motion could be accurately classified as either a fall or a non-fall using machine learning.

Keywords: Fall Detection · Depth Camera · Simulation · LSTM.

1 INTRODUCTION

Currently in Japan, there is an increasing trend of elderly people living alone, driven by the aging population [1]. Furthermore, accidents due to declining physical and cognitive functions in elderly individuals are frequently reported. Approximately 56% of these accidents are due to indoor falls in residential settings [2]. If an elderly person living alone falls indoors, and loses consciousness or is immobilized due to injury, he or she may not be able to call outside for help, which could lead to serious injury or death. Therefore, monitoring systems which can check the safety of the eldery from the outside are attracting attention [3]. Among these, fall detection in monitoring systems involves measuring the state during falls using various types of sensors. The data collected are then used for classification and decision-making through machine learning techniques. Luca et al. collected large-scale real-world fall motion data and proposed a fall detection system using accelerometers [4]. In this proposed method, acceleration during 143 fall samples was measured, and motion classification using machine learning techniques such as SVM resulted in a sensitivity of 80%. Imamura et al. proposed a fall motion classification system using Doppler sensors[5]. This method uses time-frequency spectrograms obtained from actual sensor measurements as the dataset. The motion classification method uses Long Short-Term Memory (LSTM) and achieved a detection accuracy of 93% for sudden fall movements. However, a significant issue with these methods is the substantial cost required to replicate each movement.

The purpose of our study is to develop a fall detection system for monitoring elderly individuals living alone, and to use simulation to create training data for the system. In this study, we perform simulations of falling and non-falling motions using the inverted pendulum model, which is commonly used for human motion analysis [6, 7]. For non-falling motions, stationary standing and walking motions were selected. Using the obtained time-series data, we apply machine learning to develop a fall detection system capable of classifying falling and non-falling motions.

2 OVERVIEW OF THE FALL DETECTION SYSTEM

The fall detection system proposed in this study is shown in Figure 1. This fall detection system is designed to be installed in a living space, continuously monitoring elderly individuals living alone, and capable of alerting external contacts in case of abnormal incidents such as falls. Considering 24/7 operation within a living space without the inconvenience of wearing, we selected an installed type of depth camera.

2.1 The depth camera for fall detection system

We use the Azure Kinect depth camera from Microsoft Corporation. Figure 2 shows the product's appearance and camera coordinate system. This product utilizes the Azure Kinect Body Tracking SDK provided by the same company, enabling body tracking at 30 fps. It outputs 3D skeleton coordinates of 32 body joints in millimeters, as shown in Figure 3.

2.2 Overview of fall detection method

In this study, we calculate body acceleration from skeletal coordinates obtained from Azure Kinect and the time intervals at which these coordinates were acquired. We apply machine learning using simulated acceleration data as training data to classify fall motions based on the calculated acceleration. The reason for using acceleration as a method to distinguish fall actions is to differentiate them from movements such as squatting, which resemble falls.

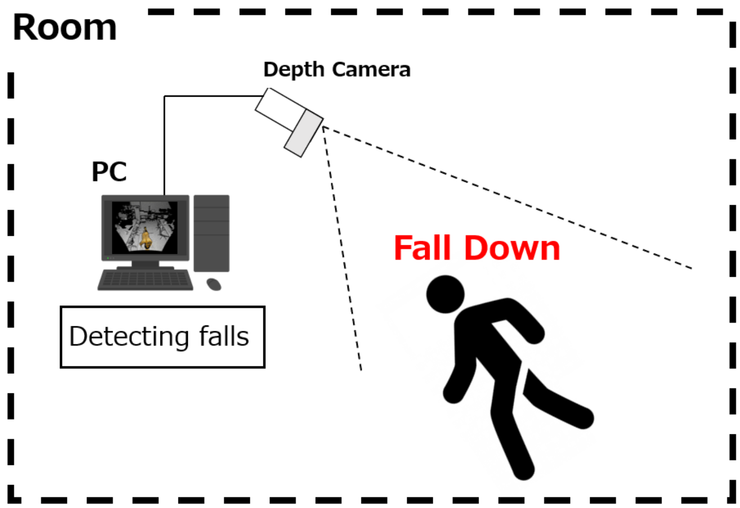


Fig. 1: Overview of fall detection system

3 CALCULATING BODY ACCELERATION USING A DEPTH CAMERA

The procedure for calculating acceleration from skeleton coordinate values obtained from Azure Kinect is described as follows.

3.1 Correction of the tilt angle at installation using rotation matrices

Azure Kinect is originally designed to be used in a horizontally mounted position. Therefore, when Azure Kinect is tilted, it cannot correctly output the

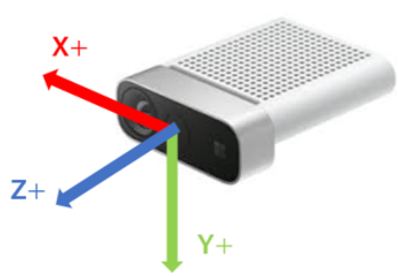


Fig. 2: Azure Kinect DK

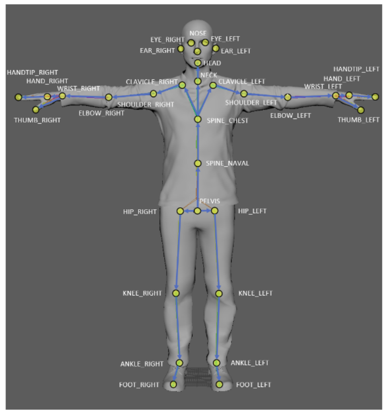


Fig. 3: Measurable joint coordinate system

skeleton coordinates. To address this, we perform a coordinate transformation around the X-axis. Figure 4 shows the overview of coordinate transformation. The coordinate transformation formula is shown in Equation 1. Y and Z represent the original coordinate axes of the Azure Kinect. θ is the camera's tilt angle. Y' and Z' are the coordinate axes after the transformation.

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{1}$$

3.2 Calculation of Body Acceleration

Figure 5 shows the flowchart for calculating acceleration from skeleton coordinates. The acceleration is derived by following this flowchart. First, upon detecting a person, obtain their skeleton coordinates and the time t at which these coordinates were captured. Next, use the transformation formula to perform a coordinate transformation, converting the value of the Y-axis to the height from the floor. Subsequently, perform a time differentiation using the obtained coordinate displacements and the time intervals at which these coordinates were captured. Then, apply a weighted moving average filter to smooth the results. Finally, calculate the acceleration by performing a time differentiation on the smoothed velocity using the time intervals at which the coordinates were captured. Similarly, apply a weighted moving average filter to smooth the calculated acceleration. Using the smoothed acceleration data for the three axes, calculate the resultant acceleration and verify the acceleration characteristics.

The weighted moving average filter is shown in Equation 2. $FilteredA$ represents the filtered value, k is the weighting constant, i is the frame number, and A denotes the velocity or acceleration being filtered.

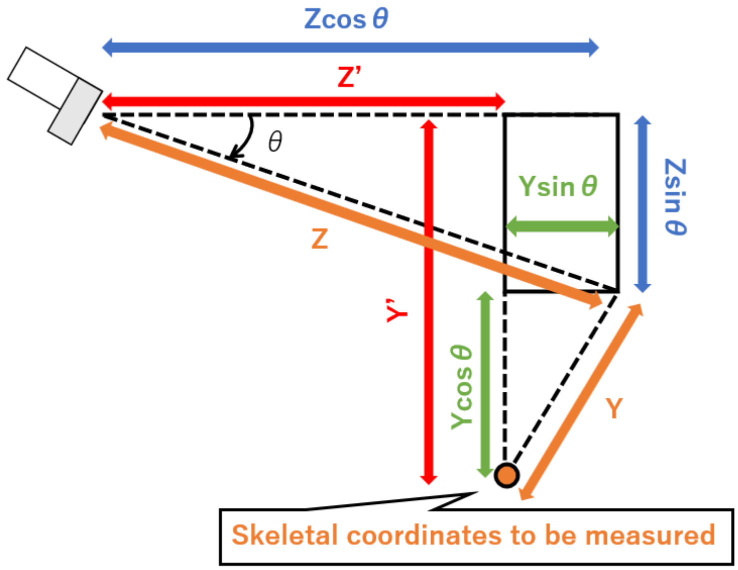


Fig. 4: Tilt angle compensation diagram for installation

$$\begin{bmatrix} FilteredA_{x_i} \\ FilteredA_{y_i} \\ FilteredA_{z_i} \end{bmatrix} = (1-k) \begin{bmatrix} FilteredA_{x_i-1} \\ FilteredA_{y_i-1} \\ FilteredA_{z_i-1} \end{bmatrix} + k \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix}, \quad k = 0.2 \tag{2}$$

4 DERIVATION OF BODY ACCELERATION

In this section, we conducted measurements of body acceleration during actual movements such as falling motions. The skeletal coordinates to be measured are the head and pelvis, as shown in Figure 3.

4.1 Measurement procedure

Figure 6 illustrates the measurement environment for the falling motion experiment. An Azure Kinect was mounted on a stand at a height of 2.1 m, angled downwards at 30°, with a mat placed in front of the stand to ensure safety. The measurements were conducted with the falling motion performed on the mat. The procedure began by positioning the subject at a stationary point approximately 4.0 m away from the Azure Kinect. After a few seconds, the subject started walking towards the mat, stumbled on the mat, and executed the falling motion. The measurements continued for a few more seconds after the fall to capture the post-fall posture before concluding the session.

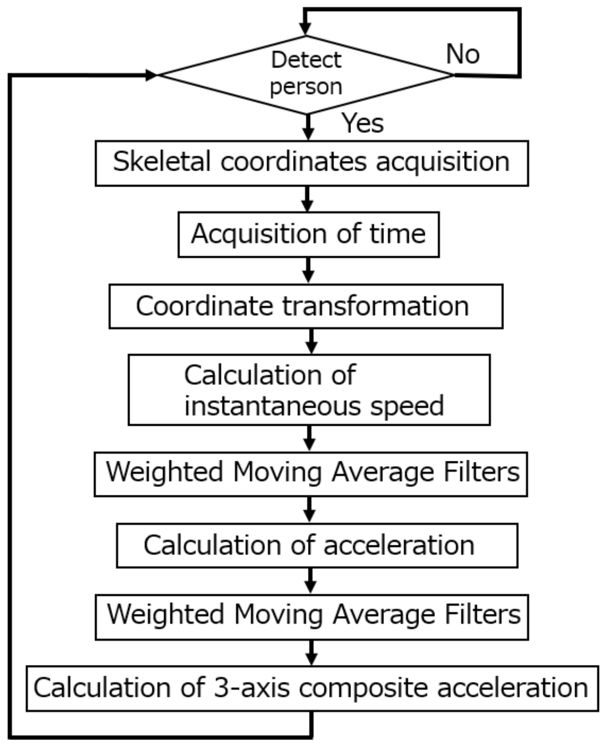


Fig. 5: Flowchart of acceleration derivation

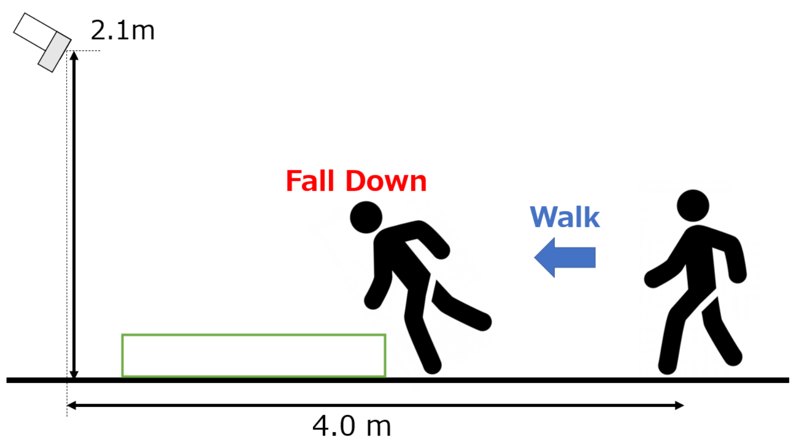
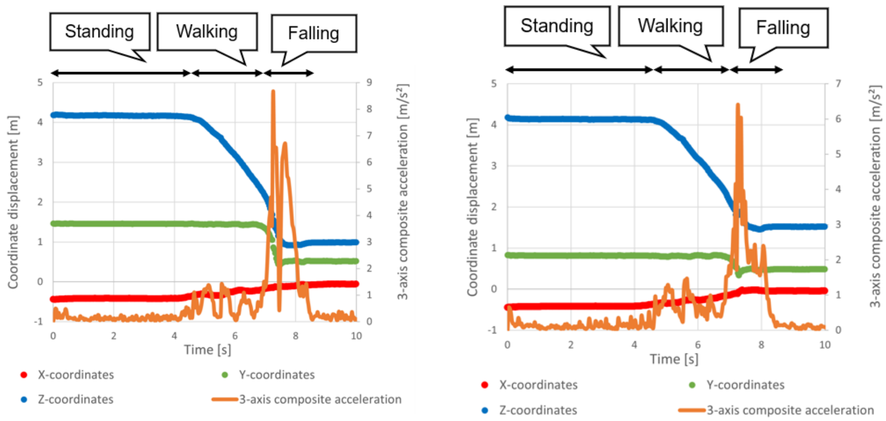


Fig. 6: Schematic of the fall experiment



(a) Head

(b) Pelvis

Fig. 7: Results of the falling experiment

4.2 Measurement results

Figure 7 shows position the changes in the X-axis, Y-axis, and Z-axis coordinates, as well as the acceleration characteristics, for the head and pelvis. The red line represents the displacement along the X-axis, the green line represents the displacement along the Y-axis, the blue line represents the displacement along the Z-axis, and the orange line represents the resultant acceleration from the three axes. The interval from 0.0 s to 4.5 s is the stationary period, from 4.5 s to 7.0 s is the walking motion period, and from 7.0 s to 7.5 s is the falling period.

It can be observed that during the stationary intervals, the resultant acceleration from the three axes remains approximately constant at 0 m/s^2. During the walking motion intervals, it can be observed that both the head and pelvis exhibit periodic increases and decreases in the resultant acceleration between 0.2 m/s^2 and 1.5 m/s^2. Furthermore, during the falling motion intervals, sudden accelerations occur at the moment of falling. The maximum acceleration observed was 8.6 m/s^2 at the head and 6.3 m/s^2 at the pelvis. From the above, it can be confirmed that sudden accelerations occur during falling movements, indicating that falling actions can be distinguished.

5 SIMULATION OF FALL AND NON-FALL MOVEMENTS USING A TRIPLE INVERTED PENDULUM

In this section, we utilized the dynamics simulator Choreonoid to create a triple inverted pendulum model and conducted simulations of falling and non-falling motions.

5.1 Development of a triple inverted pendulum

Figure 8 depicts the created inverted pendulum model. This model consists of three links and three joints. From the bottom up, the first joint simulates the ankle, the second joint simulates the knee, and the third joint simulates the hip. Additionally, from the bottom up, the first link represents the lower leg, the second link represents the thigh, and the third link represents the upper body. The main specifications of the triple inverted pendulum model created are shown in the table 1. The created inverted pendulum model has a total length of 1.5 meters and weighs 60 kilograms. It is configured with a base where three inverted pendulums are attached to rotation axis. Additionally, two models were created for fixing the ankle joint: one fixed by a rotational axis and another that operates on a cart. The model fixed by the rotational axis is used for the falling simulation, while the model fixed by the cart is used for the stationary standing and walking motion simulations. The weight of each rod was determined based on the weight ratio of the human body, ensuring that the center of gravity is located at the pelvis. Additionally, the moment of inertia for each segment was set to $I = 0.2$ kgm^2.

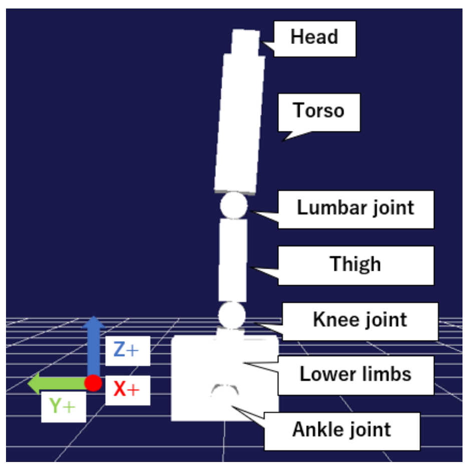


Fig. 8: Triple inverted pendulum

Table 1: Triple inverted pendulum specifications

Weight [kg]	60
Height [m]	1.5
Moment of Inertia [kgm^2]	0.2

5.2 Control Methods for the Triple Inverted Pendulum

In this section, a control method for reproducing each motion using a triple inverted pendulum is discussed. When simulating falling motions, it is necessary to ensure that each model does not interfere with each other and that the movement does not exceed the human joint range of motion. Therefore, we employed PD control, a mathematical model used for human posture control, to regulate the movements[8]. We calculated the joint input torque τ_i as shown in Equation 3 using the desired values θ , $\dot{\theta}$ for each joint angle and angular velocity, denoted as θ_i , $\dot{\theta}_i$ for the current joint angles and angular velocities. The gain constants K_p and K_d were qualitatively determined during simulation execution to ensure that the triple inverted pendulum does not exhibit erratic behavior.

In addition, control of the stationary and walking motions was performed using the choreography function of Choreonoid. The choreography function is a feature that allows specifying the joint angles and positions of the model on keyframes, automatically adjusting balance.

$$\tau_i = K_p(\theta - \theta_i) + K_d(\dot{\theta} - \dot{\theta}_i) \quad i = 1, 2, 3 \tag{3}$$

5.3 Acceleration Calculation Method for the Triple Inverted Pendulum

The following describes the method for calculating the resultant acceleration from the three axes using the coordinate values obtained from the simulation of the triple inverted pendulum. The coordinates to be obtained by the triple inverted pendulum are the head and lumbar joint from Figure 8, because the coordinates close to the head and pelvis measured by Azure Kinect were to be obtained. First, the velocity was calculated by performing a first-order differentiation on the coordinate displacements of the triple inverted pendulum. Next, using the appropriate Equation 2, the smoothed acceleration was calculated. Then, the smoothed velocity was differentiated to obtain the acceleration. Finally, the acceleration was also smoothed using the appropriate Equation 2, and the resultant acceleration was calculated from the three-axis accelerations.

5.4 Simulation of falling and non-falling scenarios using the 3-degree-of-freedom model

In this section, the results of simulating each motion are presented. The simulation timestep was set to 33.3 ms to match the sampling rate of the Azure Kinect.

Simulation of falling movements The falling motion simulation was conducted under the following conditions. The initial posture was set with angles at each joint, assuming a forward-leaning posture. No external force was applied, and the falling over was caused only by the acceleration of gravity. During the

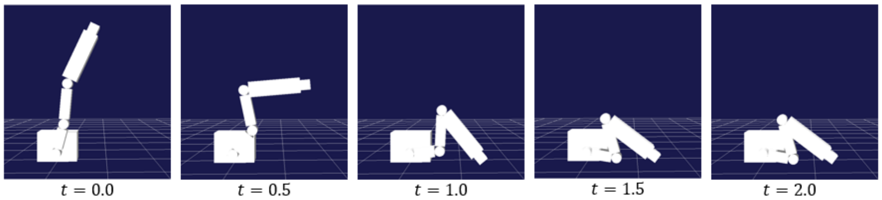
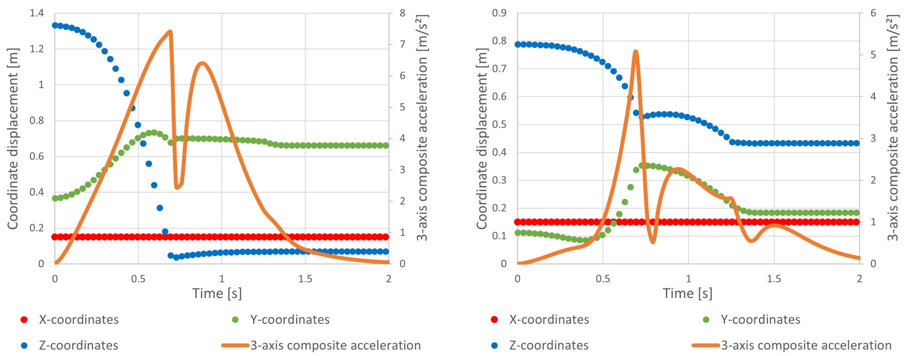


Fig. 9: Falling motion simulation



(a) Head (b) Lumbar joint

Fig. 10: Results of the falling simulation

simulation execution and termination, care was taken to ensure that each link did not interfere with others within the range of human joint mobility.

The triple inverted pendulum model during the simulation run is shown in Figure 9 and the obtained coordinate values and calculated 3-axis composite accelerations are shown in Figure 10. From Figures 9 and 10, it was observed that the acceleration increases in response to posture changes in the triple inverted pendulum model. The maximum acceleration reached was 7.4 $\mathrm{m/s^2}$ at the head coordinates and 5.0 $\mathrm{m/s^2}$ at the pelvis coordinates. From the above, it can be concluded that the triple inverted pendulum successfully generated motions that closely resemble actual falling movements.

Simulation of standing movements The standing motion simulation was conducted under the following conditions. Since a human maintains a standing posture by moving back and forth 8 mm at a rate of 4-6 times per second [9], the joint angles were adjusted so that the inverted pendulum moved back and forth 8 mm.

The triple inverted pendulum model during the simulation run is shown in Figure 11 and the obtained coordinate values and calculated 3-axis composite accelerations are shown in Figure 12. From Figure 11, it can be confirmed that

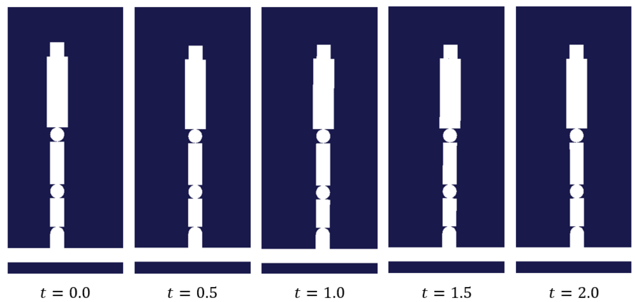
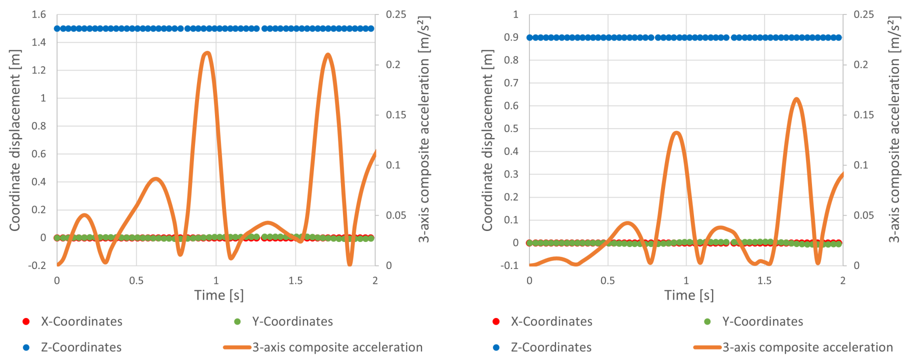


Fig. 11: Standing motion simulation



(a) Head

(b) Lumbar joint

Fig. 12: Results of the standing simulation

the triple inverted pendulum is able to maintain its posture. Additionally, from Figure 12, it can be seen that the acceleration characteristics vary from 0.0 m/s^2 to 0.2 m/s^2.

Simulation of walking movements The Walking motion simulation was conducted under the following conditions. During the human walking cycle, the stance phase, where the foot is on the ground, accounts for 60%, and the swing phase, where the foot is moved forward, accounts for 40% [9]. Therefore, the simulation was conducted to repeat the posture of the swing and stance phases every second. However, since the swing phase was fixed on the cart and the foot did not lift off the floor, only the forward swing posture was reproduced.

The triple inverted pendulum model during the simulation run is shown in Figure 13 and the obtained coordinate values and calculated 3-axis composite accelerations are shown in Figure 14. From Figure 13, it can be confirmed that each posture is being repeated and moved. Additionally, from Figure 14, it can be confirmed that a periodic acceleration characteristic similar to that of actual walking motions was obtained.

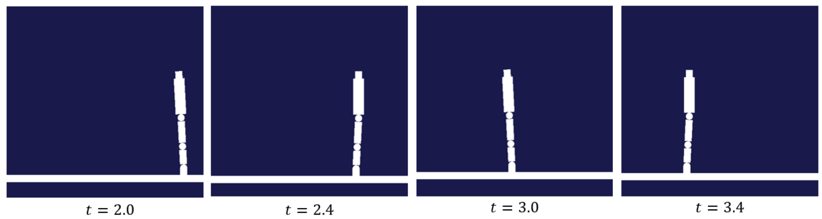
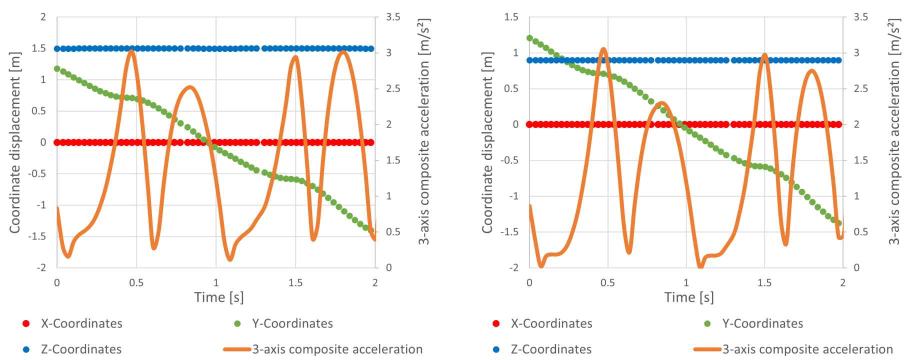


Fig. 13: Walking motion simulation



(a) Head

(b) Lumbar joint

Fig. 14: Results of the walking simulation

6 Evaluation through Machine Learning

In this section, machine learning was performed using the time series data created in Chapter 5 as training data, and the classification of each motion measured by Azure Kinect was carried out.

6.1 Creating Training Data

The training dataset used the time series data from the triple inverted pendulum model created in section 5. Each data set was processed to split the data according to the 2 s duration of the falling motion simulation. Additionally, noise was added to the created data and data inversion processing was performed, resulting in 180 samples for falling and walking motions, and 198 samples for stationary standing.

6.2 Creating Test Data

The test data consisted of time series data measured using the Azure Kinect. An example of the time series data segmentation during a fall is shown in the Figure

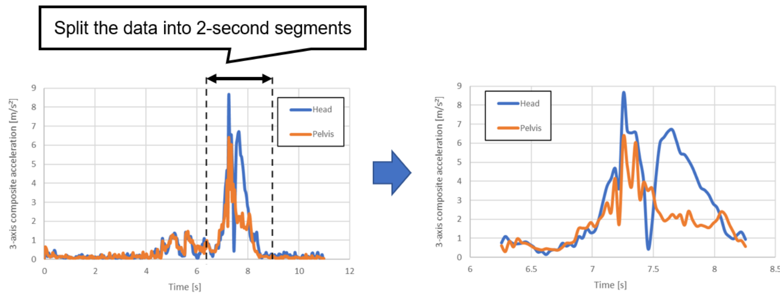


Fig. 15: Data Segmentation Example

Table 2: Hyperparameters for LSTM

Input layer dimension	2
Output layer dimension	3
Number of hidden layers	1
Dimensionality of hidden layers	100
Loss function	Cross-Entropy
Optimization function	Adam
Activation function	softmax

15. For falling motions, experiments were conducted five times in section 4, and data were processed to include a total of 2 s (1 s before and after) around the peak acceleration observed during the fall. For walking and stationary motions, each motion was performed for around 15 seconds. Data 1 second before and after the measurement was removed, and the data was processed to ensure a length of 2 seconds.

6.3 Implementation of the LSTM Model

In this study, we use LSTM(Long Short Term Memory), which is commonly employed for classification problems involving acceleration data [10, 11]. Table 2 summarizes the hyperparameters of the LSTM model. Since the input layer uses the three-axis combined acceleration of the head and torso, the dimensionality of the input layer is 2 The dimensionality of the output layer is 3, as it performs classification into three classes: falling motion, walking motion, and stationary standing.

6.4 Classification results of each motion using LSTM

Based on the parameters shown in the table 2, training was performed, and test data was loaded to classify each motion. Table 3 shows the average accuracy of five trials using the model trained with the three-axis combined acceleration

Table 3: Classification results of each motion

		Predicted value		
		Fall	Standing	Walking
True value	Fall	0.988	0.000	0.012
	Standing	0.006	0.989	0.005
	Walking	0.016	0.462	0.522

time series data measured by Azure Kinect. From Table 3, it can be confirmed that the classification accuracy for falling motion and stationary standing was approximately 99%, indicating that they were mostly correctly classified. However, the classification accuracy for walking motion was only 52%, with 46% being incorrectly classified as stationary standing.

7 CONCLUSION

In this study, to implement a fall detection system for monitoring elderly individuals living alone, we converted skeletal coordinates into acceleration and measured the acceleration characteristics during falls. From the acceleration information of the head and waist obtained during falling motions, it was confirmed that sudden acceleration occurs during a fall. Additionally, a triple inverted pendulum model simulating the human body was created in the simulator, and simulations of both falling and non-falling motions were conducted. As a result, time-series data closely resembling actual falling and non-falling motions were obtained. Finally, machine learning was performed using the time series data obtained from the simulations, which allowed for the classification of each motion.

As a future plan, we aim to generate motions similar to falling motions and make them distinguishable.

References

1. Cabinet Office, "Annual Report on the Aging Society: 2022 Edition (Complete Version), Chapter 1: The Status of Elderly People, Section 1-1, "2022, `https://www8.cao.go.jp/kourei/whitepaper/w-2022/html/zenbun/s1_1_1.html`, (2024.6.20)
2. Tokyo Fire Department, "The Reality of Daily Life Accidents as Seen from Emergency Transport Data," Disaster Prevention and Safety Division, Disaster Prevention Department, Tokyo Fire Department, 2023.
3. Y. Li, Z. Lin, Z. Huang, Z. Cai, L. Huang and Z. Wei, "A Channel Hopping LoRa Technology Based Emergency Communication System for Elderly People Living Alone," 2022 21st International Symposium on Communications and Information Technologies, pp. 19–26, 2022.
4. Luca Palmerini, Jochen Klenk, Clemens Becker, Lorenzo Chiari, "Accelerometer-Based Fall Detection Using Machine Learning: Training and Testing on Real-World Falls," Sensors (Basel), 2020.

5. T. Imamura, V. G. Moshnyaga, K. Hashimoto, “Automatic fall detection by using Doppler-radar and LSTM-based recurrent neural network,” 2022 IEEE 4th Global Conference on Life Sciences and Technologies (LifeTech), pp. 36–37, 2022.
6. Kimitaka Hase, Munkhdelger Dorjravdan, “Basic Viewpoint and Procedure of Gait Analysis,” The Japanese Journal of Rehabilitation Medicine, pp. 114–120, 2021.(In Japanese)
7. A. Kot and A. Nawrocka, “Double inverted pendulum for human hip strategy of balance modelling,” 17th International Carpathian Control Conference (ICCC), pp. 368–371, 2016.
8. Hang Zhao, Hui Zhou, Dongsheng Cai, “Simulation of Human Falling Action Considering the Movable Ranges of Body Joints,” Proceedings of the 65th National Convention of the Information Processing Society, vol.1, pp. 181–182, 2003.(In Japanese)
9. Gtz-Neumann K, Ganganalyse in der Physiotherapie, Stuttgart:, Thieme Verlag, 2016.
10. M. A. Khatun et al., , “Deep CNN-LSTM With Self-Attention Model for Human Activity Recognition Using Wearable Sensor,” IEEE Journal of Translational Engineering in Health and Medicine, vol. 10, pp. 1-16, 2022.
11. W. Ahmad, B. M. Kazmi and H. Ali “Human Activity Recognition using Multi-Head CNN followed by LSTM,” 2019 15th International Conference on Emerging Technologies (ICET), pp. 1–6, 2019.