1. INTRODUCTION

Traditional motion capture systems typically rely on specialized hardware, such as multiple high-resolution cameras and reflective markers, which can be expensive and require complex setup procedures. These systems are commonly used in industries like entertainment, biomechanics, and virtual reality to capture and analyze human movement for various purposes, including animation, sports analysis, and rehabilitation. However, the high cost and technical requirements of traditional motion capture systems limit their accessibility, particularly for smaller studios, researchers, and hobbyists. There is a growing demand for more affordable and user-friendly alternatives that can provide comparable results without compromising on accuracy.

With the rapid advancements in computer vision and machine learning, it is now possible to achieve motion capture using more affordable and ubiquitous devices, such as standard webcams. Webcam-based motion capture leverages the increasing computational power of modern computers and sophisticated algorithms to track and analyze human movement in real-time. This approach democratizes access to motion capture technology, allowing hobbyists, educators, independent developers, and researchers to utilize motion capture without the need for expensive equipment.

1.1 Objectives

The objective is to develop a methodology for capturing full-body motion using a webcam. Implement computer vision algorithms to extract relevant features from video frames, such as joint positions and limb orientations. Investigate machine learning approaches, such as pose estimation models, for accurately estimating the pose of the subject based on the extracted features. Design and implement a system for real-time reconstruction of full-body motion using the estimated pose data. Evaluate the accuracy and performance of the developed system across various scenarios and applications. Provide a cost-effective and accessible alternative to traditional motion capture systems, enabling broader adoption in industries such as animation, gaming, sports analysis, and physical therapy. By achieving these objectives, the project aims to democratize full-body motion capture technology, making it more accessible to a wider range of users and applications.

2. LITERATURE SURVEY

Wootaek Lim (2022) [2] have proposed that Markerless-can measure angular joint motions at the same level as the Vicon 3D motion capture system with markers. Because webcams are portable and consume low power, the webcam-based motion capture system will enhance the usability in natural settings, such as clinical and sports sites, beyond artificial spaces such as laboratories.

Bin Feng (2021) [1] have used the potential approach for a Marker-less motion capture system using only one camera. They tested the accuracy of their model using the Open pose algorithm and verified its functionality for gait analysis purposes.

Nobuyasu Nakano et al (2020) [3] Has examine the accuracy of 3D markerless motion capture using OpenPose with multiple video cameras through comparison with an optical marker-based motion capture. Qualitatively, 3D pose estimation using the markerless motion capture approach can correctly reproduce the movements of participants, the MAE in 80% of all trials conducted was found to be <30 mm. This relatively small error may be due to the shortcomings in the OpenPose tracking precision

Sun. K, Xiao B, Liu D., & Wang, J. (2019) [13] has examine in their" Deep High-Resolution Representation Learning for Human Pose Estimation" and proposed a high-resolution network (HRNet) for human pose estimation, which maintains high-resolution representations through the entire process and combines them with multi-resolution representations. Results and Accuracy: Achieved state-of-the-art results with an AP of 75.1% on the COCO keypoints challenge dataset.

Zhang, F (et.al) (2020) [14] "Distribution-Aware Coordinate Representation for Human Pose Estimation" proposes a novel distribution-aware coordinate representation for human pose estimation, which enhances the localization accuracy of keypoints by modeling the distribution of coordinates. Results and Accuracy: Achieved a higher accuracy than previous methods with an AP of 76.7% on the MPII Human Pose dataset.

3. MOTION CAPTURE SYSTEM

Motion capture (sometimes referred as mo-cap or mocap, for short) is the process of recording the movement of objects or people. It is used in military, entertainment, sports, medical applications, and for validation of computer vision and robots. In filmmaking and video game development, it refers to recording actions of human actors and using that information to animate digital character models in 2D or 3D computer animation. When it includes face and fingers or captures subtle expressions, it is often referred to as performance capture. In many fields, motion capture is sometimes called motion tracking, but in filmmaking and games, motion tracking usually refers more to match moving.

Fig 3.1: The motion capture System

In motion capture sessions, movements of one or more actors are sampled many times per second. Whereas early techniques used images from multiple cameras to calculate 3D positions, often the purpose of motion capture is to record only the movements of the actor, not their visual appearance. This animation data is mapped to a 3D model so that the model performs the same actions as the actor. This process may be contrasted with the older technique of rotoscoping. Camera movements can also be motion captured so that a virtual camera in the scene will pan, tilt or dolly around the stage driven by a camera operator while the actor is performing. At the same time, the motion capture system can capture the camera and props as well as the actor's performance. This allows the computer-generated characters, images and sets to have the same perspective as the video images from the camera.

A computer processes the data and displays the movements of the actor, providing the desired camera positions in terms of objects in the set. Retroactively obtaining camera movement data from the captured footage is known as match moving or camera tracking.

3.1 Methods and types of systems

Motion tracking or motion capture started as a photogrammetric analysis tool in biomechanics research in the 1970s and 1980s, and expanded into education, training, sports and recently computer animation for television, cinema, and video games as the technology matured. Since the 20th century, the performer must wear markers near each joint to identify the motion by the positions or angles between the markers. Acoustic, inertial, LED, magnetic or reflective markers, or combinations of any of these, are tracked, optimally at least two times the frequency rate of the desired motion. The resolution of the system is important in both the spatial resolution and temporal resolution as motion blur causes almost the same problems as low resolution. Since the beginning of the 21st century - and because of the rapid growth of technology - new methods have been developed. Most modern systems can extract the silhouette of the performer from the background. Afterwards all joint angles are calculated by fitting in a mathematical model into the silhouette. For movements you cannot see a change of the silhouette, there are hybrid systems available that can do both (marker and silhouette), but with less marker.

3.1.1 OPTICAL MARKER SYSTEM

Optical systems utilize data captured from image sensors to triangulate the 3D position of a subject between two or more cameras calibrated to provide overlapping projections. Data acquisition is traditionally implemented using special markers attached to an actor; however, more recent systems can generate accurate data by tracking surface features identified dynamically for each subject. Tracking many performers or expanding the capture area is accomplished by the addition of more cameras. These systems produce data with three degrees of freedom for each marker, and rotational information must be inferred from the relative orientation of three or more markers; for instance, shoulder, elbow and wrist markers providing the angle of the elbow. Newer hybrid systems are combining inertial sensors with optical sensors to reduce occlusion, increase the number of users and improve the ability to track without having to manually clean up data.

Fig 3.1.1: optimal marker-based system

3.1.2 MARKERLESS SYSTEM

Emerging techniques and research in computer vision are leading to the rapid development of the markerless approach to motion capture. Markerless systems such as those developed at Stanford University, the University of Maryland, MIT, and the Max Planck Institute, do not require subjects to wear special equipment for tracking. Special computer algorithms are designed to allow the system to analyze multiple streams of optical input and identify human forms, breaking them down into constituent parts for tracking. ESC entertainment, a subsidiary of Warner Brothers Pictures created especially to enable virtual cinematography, including photorealistic digital look-alikes for filming The Matrix Reloaded and The Matrix Revolutions movies, used a technique called Universal Capture that utilized 7 camera setup and the tracking the optical flow of all pixels over all the 2-D planes of the cameras for motion, gesture and facial expression capture leading to photorealistic results.

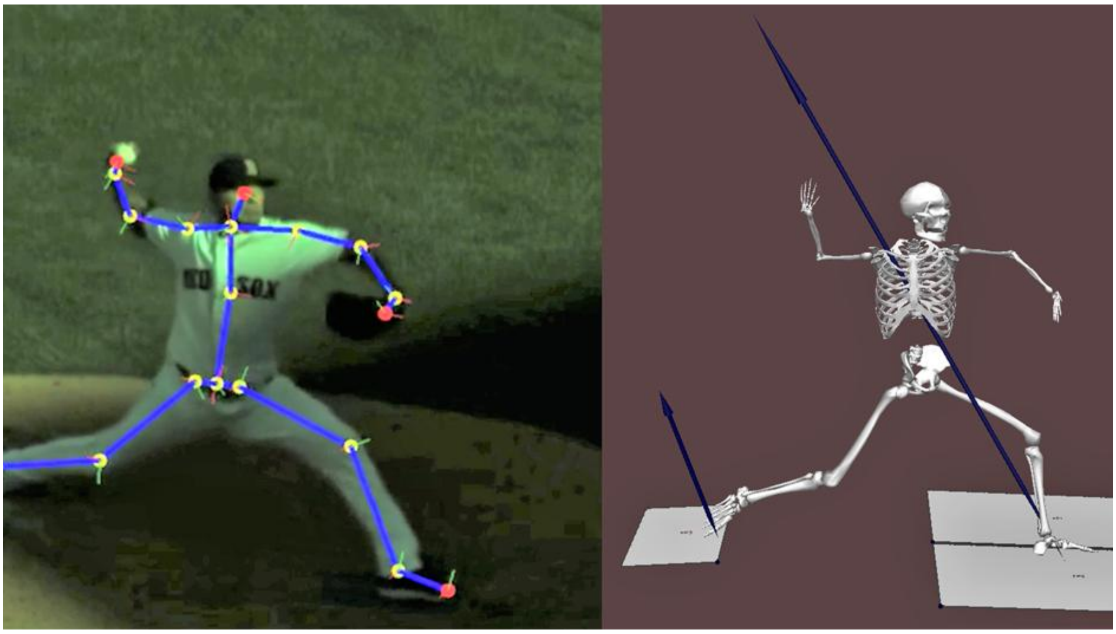

Fig 3.1.2: Markerless motion capture

3.1.3 INERTIAL SYSTEMS (NON-OPTICAL)

Inertial motion capture technology is based on miniature inertial sensors, biomechanical models and sensor fusion algorithms. The motion data of the inertial sensors (inertial guidance system) is often transmitted wirelessly to a computer, where the motion is recorded or viewed. Most inertial systems use inertial measurement units (IMUs) containing a combination of gyroscope, magnetometer, and accelerometer, to measure rotational rates. These rotations are translated to a skeleton in the software. Much like optical markers, the more IMU sensors the more natural the data. No external cameras, emitters or markers are needed for relative motions, although they are required to give the absolute position of the user if desired.

Inertial motion capture systems capture the full six degrees of freedom body motion of a human in real-time and can give limited direction information if they include a magnetic bearing sensor, although these are much lower resolution and susceptible to electromagnetic noise. Benefits of using Inertial systems include capturing in a variety of environments including tight spaces, no solving, portability, and large capture areas. Disadvantages include lower positional accuracy and positional drift which can compound over time. These systems are like the Wii controllers but are more sensitive and have greater resolution and update rates. They can accurately measure the direction to the ground to within a degree. The popularity of inertial systems is rising among game developers, mainly because of the quick and easy setup resulting in a fast pipeline.

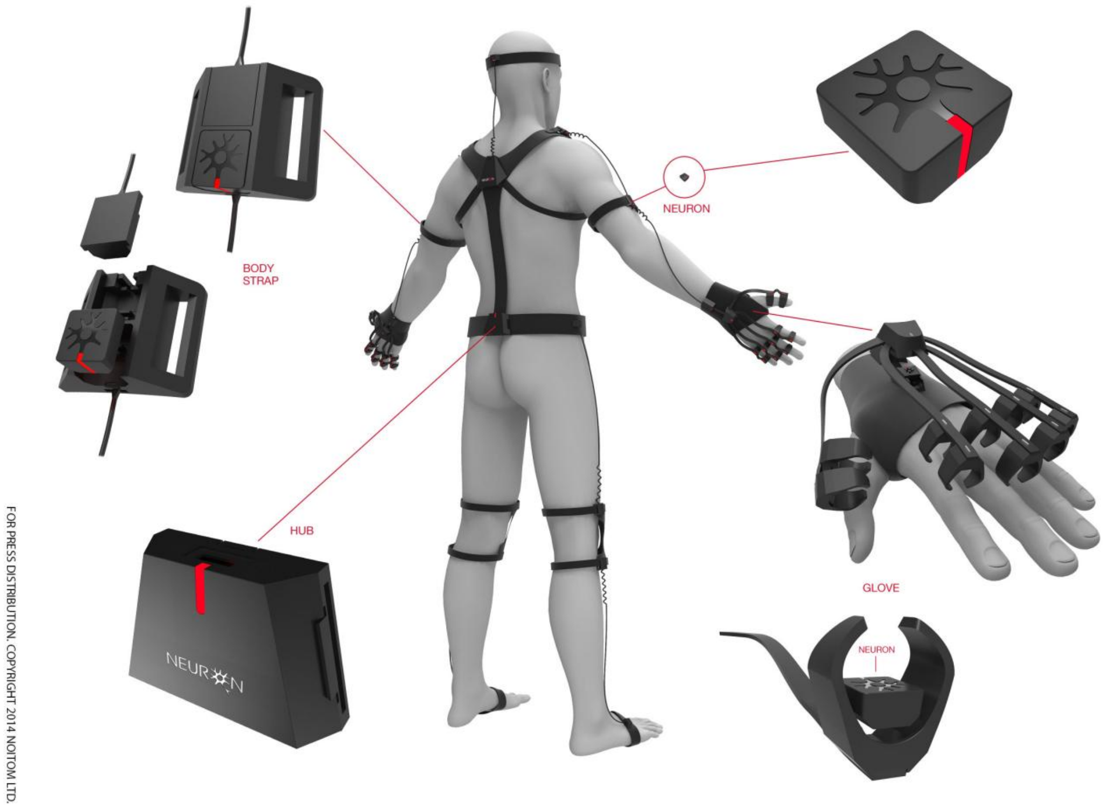

Fig 3.1.3: Inertial tracking systems

3.2 Applications of Motion Capture

1. Entertainment:

Film and Animation: Motion capture is extensively used in creating realistic animations for films and video games. Characters in movies like "Avatar" and video games like "The Last of Us" come to life through detailed mocap data.

Gaming: Game developers use mocap to create realistic character movements, enhancing the gaming experience. Titles like "FIFA" and "Madden NFL" rely heavily on mocap for player animations.

2. Sports:

Performance Analysis: Athletes use motion capture to analyze their movements, improve techniques, and prevent injuries. By studying detailed motion data, trainers can provide more effective coaching.

Rehabilitation: In physical therapy, mocap is used to monitor patient progress and tailor rehabilitation programs to individual needs.

3. Virtual and Augmented Reality:

Immersive Experiences: In VR and AR, motion capture allows users to see their own movements reflected in the virtual environment, enhancing immersion and interaction.

4. Medical Applications:

Biomechanics: Researchers use motion capture to study human movement and biomechanics, leading to advancements in prosthetics and orthotics.

Surgical Training: Mocap is employed in training simulations for surgeons, providing a realistic environment to practice procedures.

5. Human-Computer Interaction:

Gesture Recognition: Motion capture technology is used to develop gesture-based controls for interacting with computers and other devices, paving the way for more intuitive interfaces.

4. REQUIREMENT AND ANALYSIS

4.1. HARDWARE AND SOFTWARE SPECIFICATIONS

4.1.1Hardware Specifications:

Given below is the recommended hardware specification. That is required to develop the model:

1. Processor	: Intel(R) Core(TM) i5-11400H @2.70Ghz
2. Memory	: 8GB DDR4
3.Graphics Processing Unit (GPU)	: GPU with CUDA and NVIDIA Ge Force.
4. Hard disk	: 1TB of SSD

4.1.2 Software Specifications

Given below is the recommended software specification. That is required to develop the model:

Operating System : Windows 10.

Language : Python

Libraries and packages:

- Media Pipe:A popular open-source library for real-time multi-purpose key point detection and pose estimation .
- Open CV: A comprehensive computer vision library for image and video processing, including camera calibration and geometric transformations.
- Numpy : NumPy is a library for the Python programming language, adding support for large, multi - dimensional arrays and matrices.

Integrated Development Environment : Visual studio (Visual studio code editor)

By meeting these hardware and software specifications, a full-body motion capture system using webcams can achieve accurate and reliable motion tracking for various applications, including animation, sports analysis, rehabilitation, and virtual reality.

4.2 TOOLS AND TECHNOLOGIES

Motion capture using webcams involves a combination of hardware and software tools and technologies to capture, process, and analyze human movement data. Here are some of the key tools and technologies commonly used for webcam-based motion capture:

1. **Mediapipe:** MediaPipe is an open-source framework developed by Google for building cross-platform applications involving perceptual computing tasks, such as real-time hand tracking, pose estimation, facial recognition, and augmented reality effects. It provides a set of pre-trained machine learning models and a flexible pipeline for processing multimedia data, including images, video streams, and audio.

2. **OpenCV** (Open-Source Computer Vision Library): OpenCV is a powerful open-source computer vision library that provides a wide range of functions for image and video processing. It is commonly used for camera calibration, geometric transformations, and feature detection, which are essential for webcam-based motion capture.

3. **Python Programming Language**: Python is often the language of choice for developing webcam-based motion capture systems due to its simplicity, versatility, and extensive libraries for scientific computing and computer vision. Libraries like OpenCV and Mediapipe have Python bindings, making them easily accessible for developers.

4. **Webcams:**High-resolution webcams capable of capturing detailed images are essential for webcam-based motion capture. Multiple webcams may be used to capture different perspectives of the subject's movements.

5. **Computer Hardware:** A high-performance computer with sufficient processing power and memory is necessary for real-time processing of webcam footage. Multi-core processors and dedicated GPUs with CUDA support can accelerate the processing of image data.

6. **Mounting Equipment:**Sturdy tripods or mounting stands are used to securely position webcams at optimal angles for capturing the entire body. Adjustable mounts facilitate calibration and alignment of webcams.

7. 3D Reconstruction: Reconstructing 3D human poses from 2D keypoints detected by Mediapipe. These tools typically use geometric algorithms to estimate the spatial positions of joints and body parts.

By leveraging these tools and technologies, developers can create sophisticated webcam-based motion capture systems capable of accurately capturing and analyzing human movement for various applications, including animation, sports analysis, and virtual reality.

5. SYSTEM DESIGN

5.1 SYSTEM PERSPECTIVE

Full body motion capture using a standard webcam is a challenging task due to the limitations of the webcam hardware and the complexity of tracking the entire human body. However, it is possible to achieve this with certain techniques and algorithms. Here's an overview of the system perspective for full body motion capture using a webcam:

1. Camera Setup: The webcam should be positioned in a way that captures the full body of the subject from a suitable angle and distance. Good lighting conditions are essential for accurate tracking.

2. Background Subtraction: To isolate the subject from the background, background subtraction techniques are used. This involves modeling the background and then subtracting it from the current frame to obtain a foreground mask containing the subject.

3. Skeletal Tracking: Once the subject is segmented, the next step is to estimate the positions of various body joints and limbs. This can be achieved using algorithms like OpenPose, which employs deep learning models to detect and track body keypoints.

4. Pose Estimation: The detected keypoints are used to estimate the 3D pose of the subject. This involves techniques like inverse kinematics, which calculates the joint angles and positions based on the constraints of the human skeletal structure.

5. Data Representation: The tracked motion data needs to be represented in a suitable format for further processing or integration with other systems. Common formats include skeletal animation data (e.g., BVH, FBX), or marker-based motion capture data formats.

6. Real-time Processing: For real-time applications, the entire pipeline needs to be optimized for computational efficiency, often involving GPU acceleration and parallelization.

It's important to note that while webcam-based motion capture can be useful for certain applications, it has inherent limitations compared to professional motion capture systems that use multiple high-resolution cameras and specialized markers. The accuracy, robustness, and fidelity of webcam-based motion capture may not be suitable for demanding applications like professional animation or biomechanical analysis

5.2 MODULE DESCRIPTION

Full body motion capture using a webcam, the "module description" typically refers to the software or algorithmic components that enable the capture, processing, and interpretation of the motion data obtained from the webcam feed. Here's a breakdown of what such a module might include:

Camera Calibration: This module involves calibrating the webcam to ensure accurate measurements of distances and angles within the captured environment. Calibration helps correct for lens distortion and ensures that the motion captured is represented accurately.

Skeleton Tracking: This is perhaps the most crucial module in full-body motion capture. It involves identifying key points on the human body (joints, limbs, etc.) in each frame of the webcam feed. Common methods for skeleton tracking include marker-based approaches (where specific points on the body are marked with distinct markers) and markerless approaches (where computer vision algorithms identify key points without the need for markers).

Motion Reconstruction: Once the skeleton is tracked in each frame, the motion capture system reconstructs the movement of the subject in 3D space over time. This involves estimating the position and orientation of each joint at each point in time, creating a continuous representation of the subject's motion.

Gesture Recognition: Some motion capture systems include a module for recognizing specific gestures or actions performed by the subject. This could involve identifying predefined sequences of movements or recognizing gestures based on patterns in the motion data.

Performance Optimization: Depending on the application, performance optimization may be necessary to ensure real-time or near-real-time processing of the webcam feed. This could involve optimizing algorithms, leveraging parallel processing techniques, or utilizing hardware acceleration (e.g., GPU computing).

5.3 SYSTEM BLOCK DIAGRAM

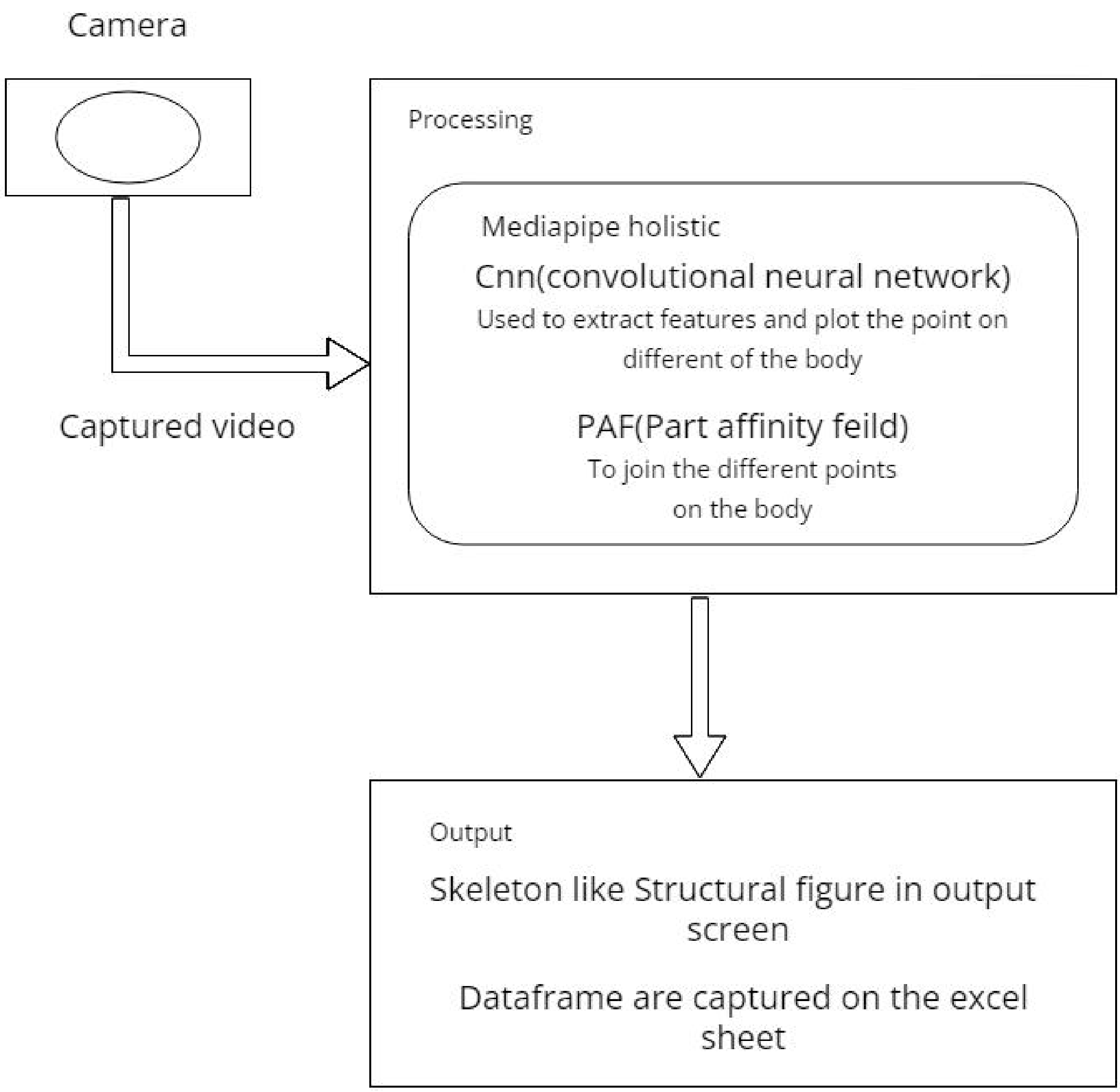

Fig 5.3.1: System block Diagram – Full body motion capture using webcam

5.4 USE CASE DIAGRAM

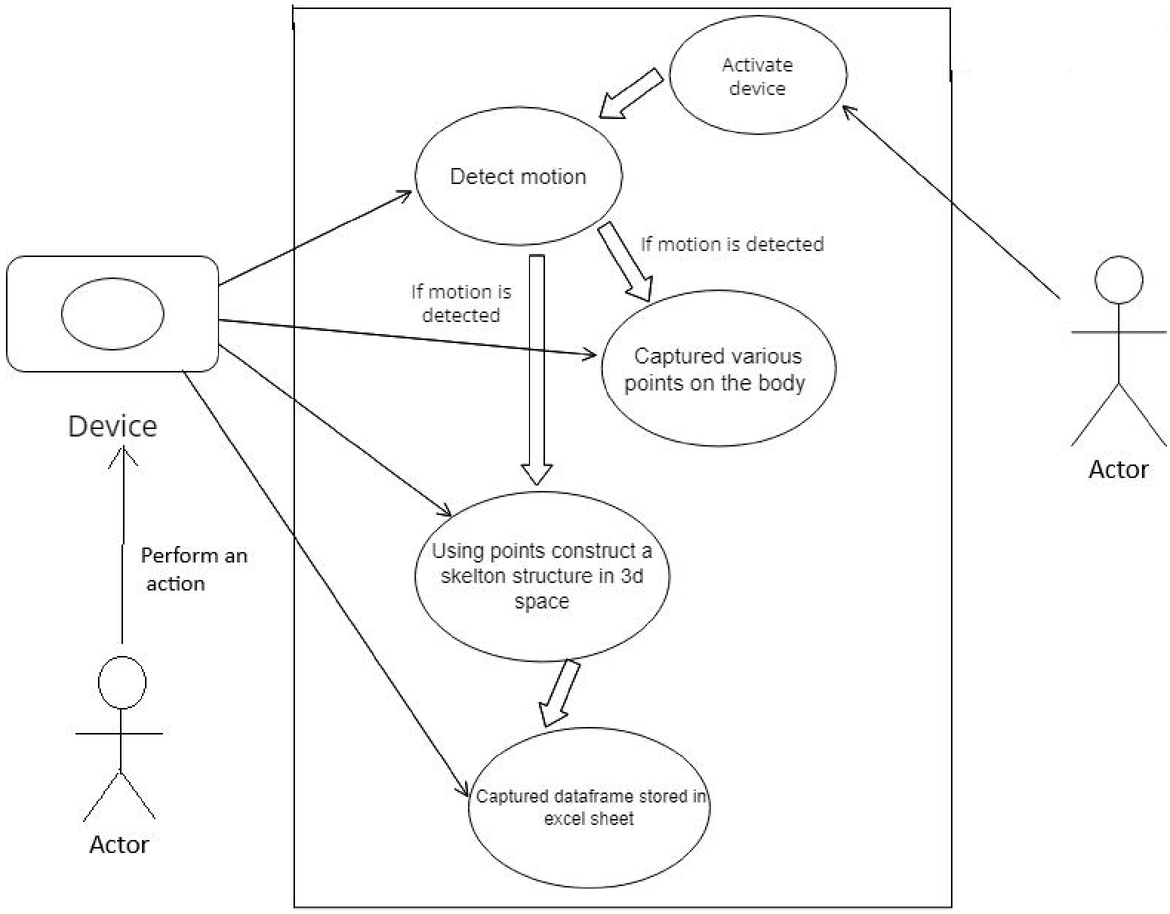

Fig 5.4.1: Use case diagram – Full body motion capture using webcam

6. IMPLEMENTATION

6.1 SOURCE CODE

```
import cv2
import mediapipe as mp
import numpy as np
import pandas as pd
import time
mp_drawing = mp.solutions.drawing_utils
mp_holistic = mp.solutions.holistic
count = 0
alldata = []
fps_time = 0
pose_tubuh = ['NOSE', 'LEFT_EYE_INNER', 'LEFT_EYE', 'LEFT_EYE_OUTER',
'RIGHT_EYE_INNER', 'RIGHT_EYE', 'RIGHT_EYE_OUTER',
            'LEFT_EAR', 'RIGHT_EAR', 'MOUTH_LEFT', 'MOUTH_RIGHT',
            'LEFT_SHOULDER', 'RIGHT_SHOULDER', 'LEFT_ELBOW',
'RIGHT_ELBOW', 'LEFT_WRIST', 'RIGHT_WRIST', 'LEFT_PINKY',
            'RIGHT_PINKY', 'LEFT_INDEX', 'RIGHT_INDEX', 'LEFT_THUMB',
            'RIGHT_THUMB', 'LEFT_HIP', 'RIGHT_HIP', 'LEFT_KNEE', 'RIGHT_KNEE',
'LEFT_ANKLE', 'RIGHT_ANKLE',
            'LEFT_HEEL', 'RIGHT_HEEL', 'LEFT_FOOT_INDEX',
'RIGHT_FOOT_INDEX']
pose_tangan = ['WRIST', 'THUMB_CPC', 'THUMB_MCP', 'THUMB_IP', 'THUMB_TIP',
'INDEX_FINGER_MCP', 'INDEX_FINGER_PIP',
             'INDEX_FINGER_DIP', 'INDEX_FINGER_TIP', 'MIDDLE_FINGER_MCP',
             'MIDDLE_FINGER_PIP', 'MIDDLE_FINGER_DIP', 'MIDDLE_FINGER_TIP',
'RING_FINGER_PIP', 'RING_FINGER_DIP',
             'RING_FINGER_TIP',
             'RING_FINGER_MCP', 'PINKY_MCP', 'PINKY_PIP', 'PINKY_DIP',
'PINKY_TIP']
```

```
pose_tangan_2 = ['WRIST2', 'THUMB_CPC2', 'THUMB_MCP2', 'THUMB_IP2',
'THUMB_TIP2', 'INDEX_FINGER_MCP2',
              'INDEX_FINGER_PIP2', 'INDEX_FINGER_DIP2', 'INDEX_FINGER_TIP2',
'MIDDLE_FINGER_MCP2',
              'MIDDLE_FINGER_PIP2', 'MIDDLE_FINGER_DIP2',
'MIDDLE_FINGER_TIP2', 'RING_FINGER_PIP2',
              'RING_FINGER_DIP2', 'RING_FINGER_TIP2',
              'RING_FINGER_MCP2', 'PINKY_MCP2', 'PINKY_PIP2', 'PINKY_DIP2',
'PINKY_TIP2']
cap = cv2.VideoCapture("I:¥motion capture¥¥jj1_nvKjgVcZ.mp4")
# suc,frame_video = cap.read()
# vid_writer = cv2.VideoWriter('pose.avi', cv2.VideoWriter_fourcc('M','J','P','G'), 10,
(frame_video.shape[1], frame_video.shape[0]))
with mp_holistic.Holistic(
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5) as holistic:
  while cap.isOpened():
    success, image = cap.read()
    if not success:
      print("Ignoring empty camera frame.")
      # If loading a video, use 'break' instead of 'continue'.
      continue
    # Flip the image horizontally for a later selfie-view display, and convert
    # the BGR image to RGB.
    image = cv2.cvtColor(cv2.flip(image, 1), cv2.COLOR_BGR2RGB)

 # To improve performance, optionally mark the image as not writeable to
    # pass by reference.
    image.flags.writeable = False
    results = holistic.process(image)
    # Draw landmark annotation on the image.
    image.flags.writeable = False
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
```

```
    image_output = np.copy(image)
    image = np.zeros(image.shape)
    mp_drawing.draw_landmarks(
        image, results.left_hand_landmarks, mp_holistic.HAND_CONNECTIONS)
    mp_drawing.draw_landmarks(
        image, results.right_hand_landmarks, mp_holistic.HAND_CONNECTIONS)
    mp_drawing.draw_landmarks(
        image, results.pose_landmarks, mp_holistic.POSE_CONNECTIONS)
    #  if(results.pose_landmarks is not None and results.left_hand_landmarks is not None
and results.right_hand_landmarks is not None):
    if results.pose_landmarks:
        data_tubuh = {}
        for i in range(len(pose_tubuh)):
            results.pose_landmarks.landmark[i].x = results.pose_landmarks.landmark[i].x *
image.shape[0]
            results.pose_landmarks.landmark[i].y = results.pose_landmarks.landmark[i].y *
image.shape[1]
            data_tubuh.update(
                {pose_tubuh[i]: results.pose_landmarks.landmark[i]}
            )
        alldata.append(data_tubuh)
    """if results.right_hand_landmarks:
        data_tangan_kanan = {}
        for i in range(len(pose_tangan)):
            results.right_hand_landmarks.landmark[i].x =
results.right_hand_landmarks.landmark[i].x * image.shape[0]

            results.right_hand_landmarks.landmark[i].y =
results.right_hand_landmarks.landmark[i].y * image.shape[1]
            data_tubuh.update(
                {pose_tangan[i]: results.right_hand_landmarks.landmark[i]}
            )
        alldata.append(data_tubuh)"""
```

```
        """if results.left_hand_landmarks:
            data_tangan_kiri = {}
            for i in range(len(pose_tangan)):
                results.left_hand_landmarks.landmark[i].x =
results.left_hand_landmarks.landmark[i].x * image.shape[0]
                results.left_hand_landmarks.landmark[i].y =
results.left_hand_landmarks.landmark[i].y * image.shape[1]
                data_tubuh.update(
                    {pose_tangan_2[i]: results.left_hand_landmarks.landmark[i]}
                )
            alldata.append(data_tubuh)"""
        # cv2.namedWindow('MediaPipe Holistic', cv2.WND_PROP_FULLSCREEN)
        # cv2.setWindowProperty('MediaPipe Holistic', cv2.WND_PROP_FULLSCREEN,
cv2.WINDOW_FULLSCREEN)
        cv2.putText(image, "FPS: %f" % (1.0 / (time.time() - fps_time)), (10, 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.5,
                    (0, 255, 0), 2, )
        cv2.imshow('MediaPipe Holistic', image)  # sudah menampilkan backgrounnd hitam
dan skeleton
        cv2.imshow('Skeleton structure', image_output)
        count = count + 1
        print(count)
        fps_time = time.time()
        # vid_writer.write(image)
        # plt.imshow((image*255).astype(np.uint8))
        # plt.savefig("image-frame/" + str(count) + ".jpg")
        if cv2.waitKey(5) & 0xFF == ord('q'):

    df = pd.DataFrame(alldata)
            df.to_excel('I:/motion capture/coordinator2.xlsx')
            break
cap.release()
cv2.destroyAllWindows()
```

7. RESULTS

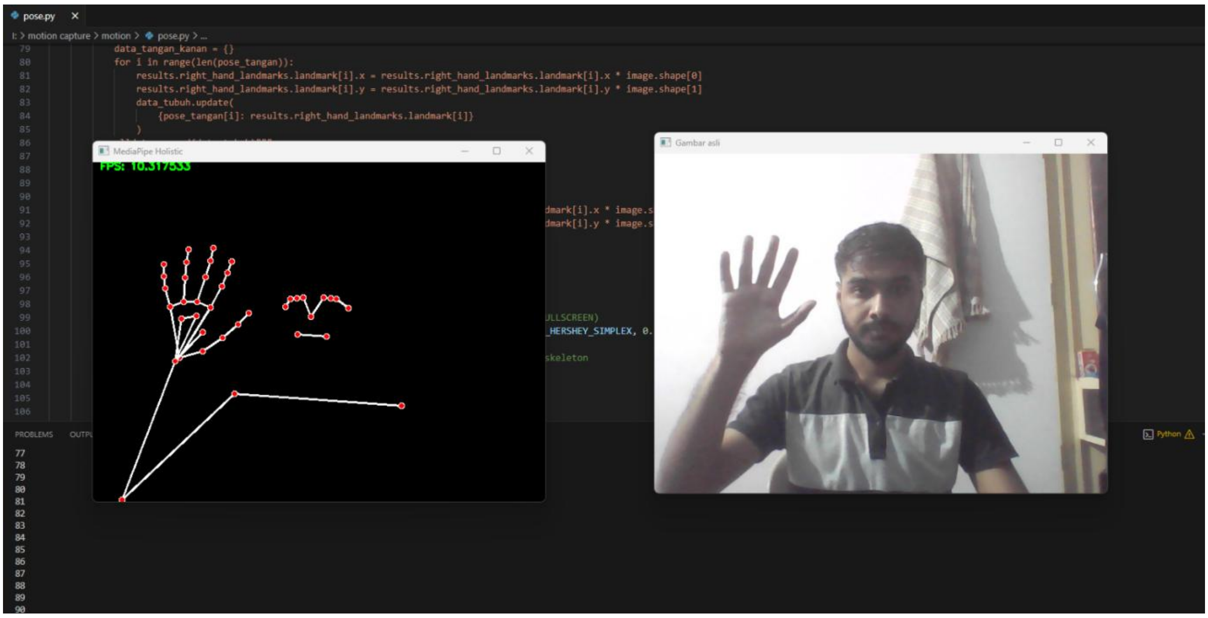

Fig 7.1.1 This figure shows real time landmarks capture using webcam

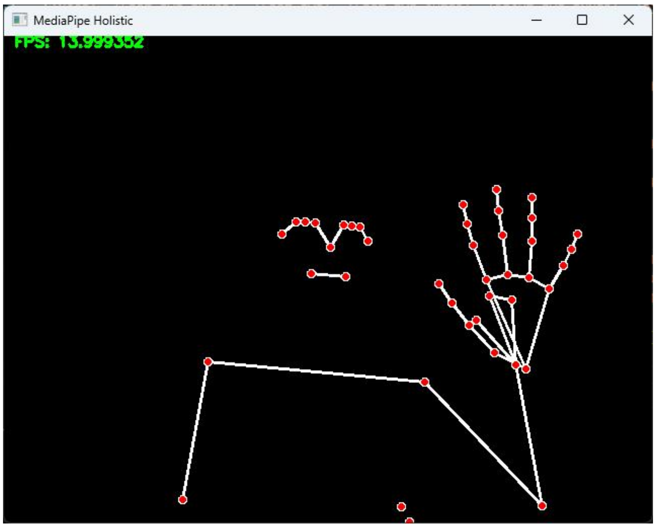

Fig 7.1.2: This window captured landmarks, which is showing the pose landmarks as well as the Hand landmarks in 3-demension

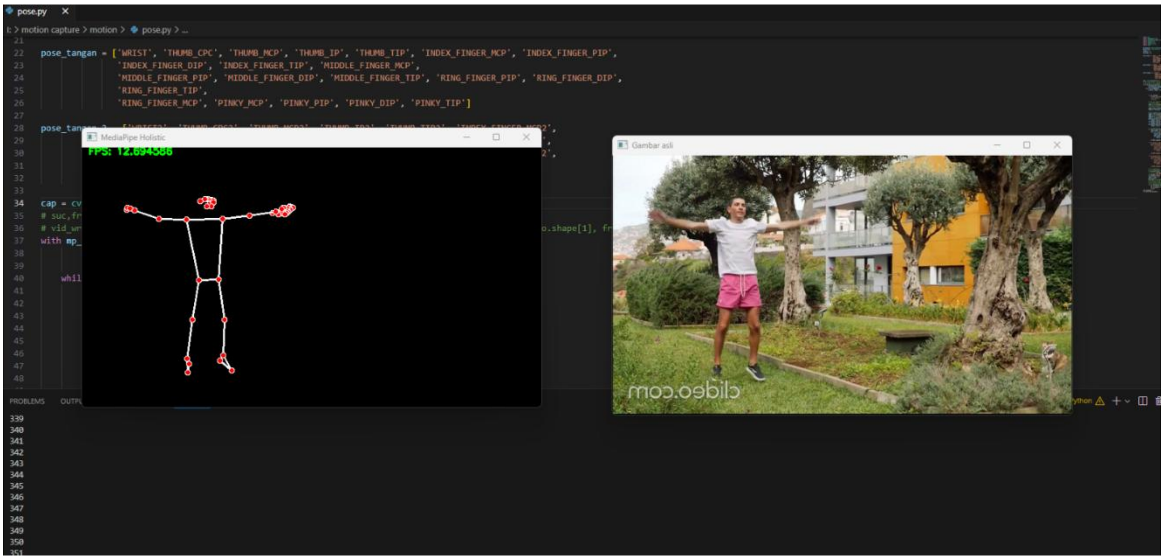

Fig 7.1.3: Shows the result of video feed and on the left side captured coordinates without video along with frame rate.

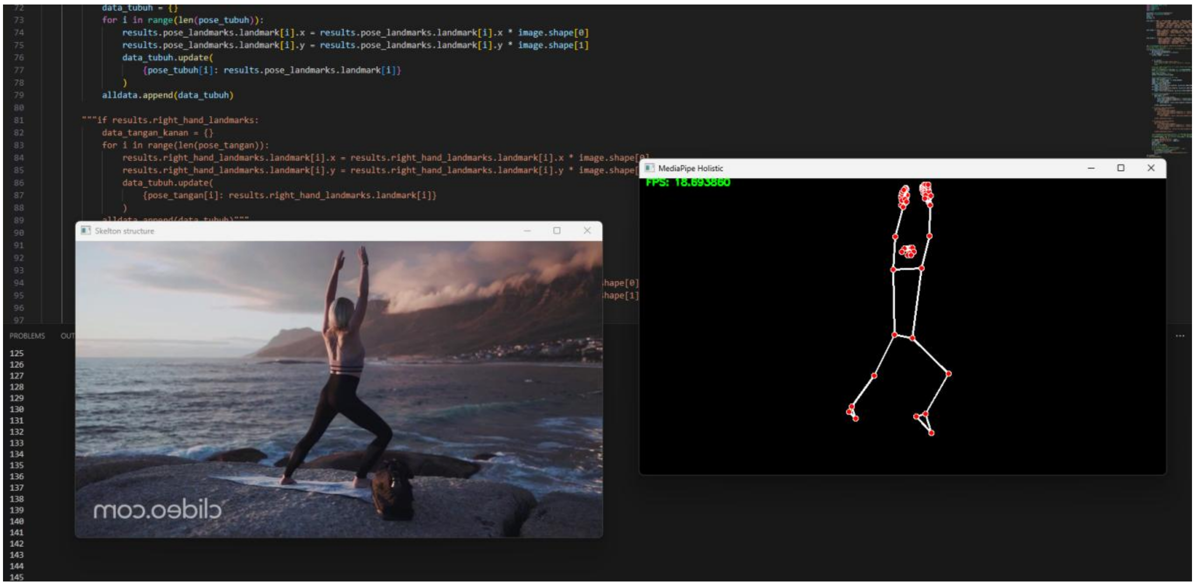

Fig7.1.4: Another video frames capturing the coordinates of the person and display on the other output screen

L21

x: 180.64091
y: 168.37778
z: -0.22402215

	A	B	C	D	E	F	G	H	I	J
1		NOSE	T_EYE_INN	LEFT_EYE	T_EYE_OU	HT_EYE_IN	RIGHT_EYE	HT_EYE_OU	LEFT_EAR	RIGHT_EAR
2	0	x: 181.454	x: 181.788	x: 182.165	x: 182.557	x: 180.778	x: 180.418	x: 179.980	x: 183.663	x: 179.503
3	1	x: 181.438	x: 181.868	x: 182.359	x: 182.896	x: 180.716	x: 180.222	x: 179.594	x: 184.107	x: 178.797
4	2	x: 181.399	x: 181.899	x: 182.439	x: 183.040	x: 180.629	x: 180.045	x: 179.330	x: 184.406	x: 178.575
5	3	x: 181.365	x: 181.988	x: 182.614	x: 183.328	x: 180.562	x: 179.884	x: 179.068	x: 184.835	x: 178.185
6	4	x: 181.403	x: 182.293	x: 183.049	x: 183.918	x: 180.559	x: 179.829	x: 178.954	x: 185.389	x: 177.888
7	5	x: 181.663	x: 182.814	x: 183.719	x: 184.683	x: 180.596	x: 179.830	x: 178.942	x: 186.000	x: 177.698
8	6	x: 181.771	x: 182.997	x: 183.935	x: 184.918	x: 180.599	x: 179.811	x: 178.889	x: 186.181	x: 177.515
9	7	x: 181.861	x: 183.170	x: 184.091	x: 185.044	x: 180.633	x: 179.818	x: 178.895	x: 186.240	x: 177.468
10	8	x: 181.926	x: 183.301	x: 184.270	x: 185.223	x: 180.637	x: 179.797	x: 178.834	x: 186.487	x: 177.349
11	9	x: 181.970	x: 183.430	x: 184.420	x: 185.374	x: 180.642	x: 179.778	x: 178.781	x: 186.648	x: 177.319
12	10	x: 182.088	x: 183.657	x: 184.674	x: 185.607	x: 180.721	x: 179.789	x: 178.784	x: 186.869	x: 177.319
13	11	x: 182.343	x: 184.024	x: 185.030	x: 185.949	x: 180.939	x: 179.889	x: 178.822	x: 187.190	x: 177.332
14	12	x: 182.582	x: 184.249	x: 185.219	x: 186.099	x: 181.263	x: 180.177	x: 179.052	x: 187.389	x: 177.359

Fig7.1.5: The captured data frame is stored in the form of excel sheet

8. CONCLUSION

8.1 CONCLUSION

In this study, we can see that webcam-based full-body motion capture systems offer a versatile and cost-effective solution for capturing and analysing human movement, with potential applications in various fields ranging from entertainment and gaming to healthcare and sports science. As technology continues to advance, these systems are expected to become even more accessible and capable, opening new possibilities for interactive and immersive experiences.

Through this project that webcam mocap is a good option for basic animations or beginners. If you need high precision mocap for professional projects, traditional suits are the way to go. There are some additional points that we can consider; Some software offers better accuracy with multiple webcams. New AI-powered tools are improving webcam mocap capabilities. For best results, ensure good lighting, wear tight clothing with contrasting colors, and maintain a clean background.

8.2 FUTURE ENHANCEMENT

Enhancements for full-body motion capture using webcams could significantly improve the quality and accessibility of motion capture technology. Here are some potential future enhancements:

Integration with AR and VR: Integrate motion capture technology with augmented reality (AR) and virtual reality (VR) systems, allowing users to interact naturally with virtual objects and environments.

Improved Accuracy: Enhance the accuracy of motion capture algorithms to better detect subtle movements and gestures, enabling more precise control and representation of body movements.

Multi-person Tracking: Enable the system to track multiple individuals simultaneously, facilitating group activities, multiplayer gaming, and collaborative virtual environments.

Robustness to Environmental Factors: Develop techniques to make the motion capture system robust to variations in lighting conditions, background clutter, and occlusions, ensuring reliable performance in diverse environments.

Integration with AI Assistants: Integrate motion capture technology with AI assistants to enable hands-free control of devices and applications through natural body movements and gestures.

Biometric Analysis Incorporate biometric analysis capabilities to extract additional information from body movements, such as emotional states, physical fatigue, or health-related indicators, for applications in healthcare, sports training, and human-computer interaction.

Privacy and Security Measures: Implement privacy-preserving techniques to ensure that sensitive user data captured by the motion capture system is securely handled and protected from unauthorized access or misuse.

Cross-platform Compatibility: Ensure compatibility with a wide range of devices and operating systems, allowing users to use motion capture technology on smartphones, tablets, PCs, and other devices.

Facial Expression Recognition: Integrate facial expression recognition technology to capture and analyze facial movements, enabling more realistic avatars and emotional expression in virtual environments.

9. REFERENCES / BIBLIOGRAPHY

1. Bin Feng "Marker-less Motion Capture System Using OpenPose" December 2021.
2. Wootaek Lim, "Markerless Motion capture system based on webcam Using OpenPose", September 2022.
3. Nobuyasu Nakano (et.al) "Evaluation of 3D Markerless Motion Capture Accuracy Using OpenPose with Multiple Video Cameras" May 2020.
4. Real-time Full Body Webcam Mocap in Unreal Engine - Dollars MONO Tutorial by Duzgen (http://www.youtube.com/watch?v=ayTm856fDBQ)
5. Webcam Motion Capture Setup by Cooki Kunai (http://www.youtube.com/watch?v=983QawjOyhc)
6. Real-Time Motion Capture with a Cheap Webcam (or iPhone): TDPT - Unreal Engine 5 Tutorial by Duzgen (http://www.youtube.com/watch?v=BlMd-LQGVyI)
7. Markerless Fullbody Motion Capture using Only a Single Webcam! by Webcam Motion Capture (http://www.youtube.com/watch?v=b6IhHXF3nx8)
8. VR Full Body Tracking for Everyone Using Webcams! - MocapForAll Tutorialby The Mysticle (http://www.youtube.com/watch?v=vIsgH-2XGeM)
9. Motion capture system description (https://iptif.tech/motion-capture-system/).
10. Optical marker-based system (https://3dcoil.grupopremo.com/blog/tracking-systems-virtual-reality-the-best-choice/).
11. Marker less system (https://news.miami.edu/edu/stories/2020/09/kinatrax-donates-advanced-motion-analysis-system-to-orovitz-laboratories,-bringing-ai-to-biomechanics-research.html)
12. Inertial tracking system (https://cgicoffee.com/blog/2017/04/first-production-motion-capture-session-report).
13. Sun, Ke (et. al) Deep High-Resolution Representation Learning for Human Pose Estimation (2019).
14. Zhang, F.(et.al) "Distribution-Aware Coordinate Representation for Human Pose Estimation (2020)