PoseIQ™: Automated Motion Capture and Analysis on the Go

Dr. Hossein Mokhtarzadeh, PhD
https://poseiq.com/
Explore the PoseIQ App and try PoseIQ2Colab!

Abstract

Motion capture (MoCap) technology has long been a cornerstone for biomechanical analysis, sports science, and rehabilitation. However, traditional MoCap systems require specialized hardware and complex workflows. PoseIQ introduces a streamlined, browser-based solution leveraging MediaPipe's pose detection technology to provide an automated, user-friendly pipeline from data capture to analysis. PoseIQ™ enables users to conduct motion experiments, process data, and refine analyses iteratively in Google Colab—all in a few simple steps. This paper describes the PoseIQ™ methodology for MoCap and its application as a robust, accessible tool for motion analysis.

Introduction

PoseIQ bridges the gap between traditional motion capture systems and accessible, web-based solutions. While this paper focuses on the integration of PoseIQ with Google Colab as an example, the broader aim of PoseIQ™ is to serve a global community with versatile motion analysis tools, including video-based live analytics. This particular implementation highlights its potential for researchers and practitioners in human movement science, showcasing how PoseIQ can streamline motion data collection and analysis.

Methodology

System Workflow

PoseIQ simplifies the motion analysis pipeline into two key loops (Fig. 1):

1. **Experiment Loop:** Users capture motion data using their webcam, process it in real-time to calculate pose landmarks and joint angles, and save the data as a JSON file.
2. **Analysis Loop:** Users upload the JSON data to Google Colab, analyze it using pre-written Python code, and refine the analysis until satisfactory results are achieved.

The system allows for iterative experimentation, enabling users to adjust and retest as needed. The workflow is illustrated in the figure below:

Step-by-Step Workflow

1. Experiment Loop

- **Webcam Capture:** Users open the PoseIQ app in their browser to record motion using their webcam.
- **Pose Landmarks and Joint Angles:** MediaPipe detects pose landmarks in real-time, and joint angles are calculated for key joints such as elbows, knees, and shoulders.
- **Data Export:** The recorded data is saved as a JSON file containing pose landmarks and joint angles. Our future work will incorporate diverse biomechanical and human movement data.

2. Analysis Loop

The analysis loop focuses on processing the captured data and refining results iteratively.

- **Data Upload:** The JSON file is uploaded to Google Colab.
- **Pre-Written Analysis Code:** A provided Python script processes the data, creating visualizations such as joint angle plots and statistical summaries.
- **Refinement:** Users can iteratively refine their analysis in Colab or return to the experiment loop to collect better data if needed.

Advantages of PoseIQ™

PoseIQ is a versatile and powerful tool for motion capture and analysis. It requires only a standard webcam and browser, making it highly accessible without the need for specialized hardware. The system automates data capture, processing, and analysis with minimal user intervention, supporting experimentation and refinement through clearly defined iterative loops. Its portability enables users to perform MoCap analysis anywhere with an internet connection. Additionally, PoseIQ is scalable, allowing integration with advanced machine learning models for future near real-time feedback within Colab.

Applications

PoseIQ™ has diverse applications across various fields. It enables the analysis of athletic performance and joint biomechanics in sports science, supports monitoring patient progress during physiotherapy in rehabilitation, facilitates teaching fundamental biomechanical concepts interactively in education, and allows researchers to conduct lightweight motion studies without the need for specialized equipment.

Conclusion

PoseIQ offers an accessible, automated, and iterative solution for motion capture and analysis. By leveraging modern browser-based technologies, PoseIQ™ empowers users to conduct motion studies with minimal setup, making MoCap technology available to a broader audience. Future iterations will include real-time feedback, cloud integration, and multi-user capabilities, further expanding its potential.

Acknowledgments

PoseIQ™ is built upon open-source technologies, including MediaPipe for pose detection, Mermaid Live for visualizing workflows, and tools like Google Colab for cloud-based analysis. Special thanks to OpenAI for facilitating conceptual development and brainstorming. These contributions have been instrumental in realizing a simple, scalable, and accessible motion capture solution.

Keywords: Motion Capture, Pose Detection, MediaPipe, Google Colab, Biomechanics, Automated Analysis

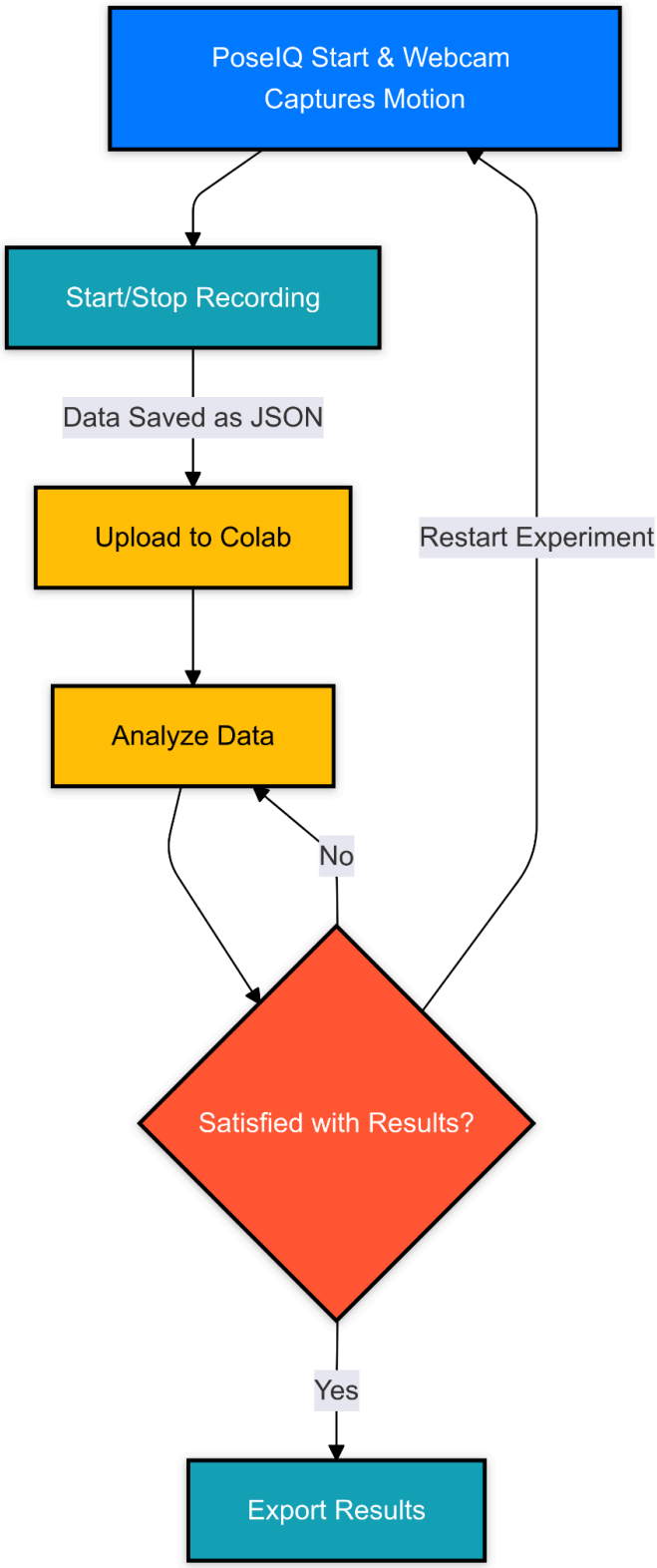

Figure 1. Workflow of PoseIQ™ showing the iterative loops for experimentation and analysis.

Appendix: Step-by-Step Instructions for Google Colab

Recording Complete!

Your pose data has been downloaded as `pose_data.json`.

Next Steps:

1. Go to Google Colab.
2. Create a new notebook by clicking on **File > New Notebook**.
3. In the new notebook, click on **Runtime > Change runtime type** and ensure that the runtime is set to Python 3.
4. Copy and paste the following Python code into a cell in your Colab notebook:

```
import json
import pandas as pd
import matplotlib.pyplot as plt
from google.colab import files

# Step 1: Upload the JSON file
print("Please upload your 'pose_data.json' file.")
uploaded = files.upload()

# Step 2: Load the JSON data
if 'pose_data.json' in uploaded:
    with open('pose_data.json', 'r') as f:
        data = json.load(f)
    print("JSON data loaded successfully.")
else:
    # If the file name is different, adjust accordingly
    file_name = list(uploaded.keys())[0]
    with open(file_name, 'r') as f:
        data = json.load(f)
    print(f"JSON data loaded successfully from {file_name}.")

# Step 3: Convert JSON data to DataFrame
landmarks = []
angles = []
for entry in data:
    timestamp = entry.get('timestamp')
    for i, landmark in enumerate(entry.get('landmarks', [])):
        landmark_entry = {
            'id': i,
            'timestamp': timestamp,
            'x': landmark.get('x'),
            'y': landmark.get('y'),
            'z': landmark.get('z'),
            'visibility': landmark.get('visibility')
        }
        landmarks.append(landmark_entry)
    angles.append(entry.get('angles'))

df_landmarks = pd.DataFrame(landmarks)
df_angles = pd.DataFrame(angles)
print("Landmarks DataFrame:")
display(df_landmarks.head())
```

```
print("Angles DataFrame:")
display(df_angles.head())

# Step 4: Plot Joint Angles Over Time
plt.figure(figsize=(12, 8))
for angle in df_angles.columns:
    plt.plot(df_angles.index, df_angles[angle], label=angle.replace('_', ' ').title())

plt.title('Joint Angles Over Time')
plt.xlabel('Frame')
plt.ylabel('Angle (degrees)')
plt.legend()
plt.grid(True)
plt.show()

# Step 5: Optional - Save Processed Data as CSV
# Uncomment the lines below to save and download the processed data
# df_landmarks.to_csv('pose_landmarks.csv', index=False)
# df_angles.to_csv('pose_angles.csv', index=False)
# files.download('pose_landmarks.csv')
# files.download('pose_angles.csv')
```