applied sciences

MDPI

Article

A Time-Scalable Posture Detection Algorithm for Paraplegic Patient Rehabilitation Using Exoskeleton-Type Wearable Robots

Ho-Won Lee, Kyung-Oh Lee, Yoon-Jae Chae, Se-Yeob Kim and Yoon-Yong Park *

Department of Computer and Electronics Convergence Engineering, Sunmoon University, Asan-si 31460, Korea; kck4156@gmail.com (H.-W.L.); leeko@sunmoon.ac.kr (K.-O.L.); kodeep@sunmoon.ac.kr (Y.-J.C.); kimseysob@gmail.com (S.-Y.K.)
* Correspondence: yypark@sunmoon.ac.kr; Tel.: +82-10-8803-2365

Abstract: Traditionally, paraplegic patients have relied on a wheelchair to travel. However, new developments in walking assistance technology have led to promising exoskeleton-type wearable robots that can help paraplegic patients walk. Operation of this new robotic device requires that patients have appropriate training to ensure safe and optimal use. Here, we propose an algorithm that can optimize rehabilitation outcomes by comparing posture data generated during the rehabilitation of a paraplegic patient wearing a body-tracking sensor with reference posture data. The proposed algorithm guarantees a certain level of accuracy when comparing rehabilitation and reference posture data. It can also correct for timescale differences between reference and rehabilitation data to ensure a high level of accuracy. Compared with other algorithms that perform similar functions, this algorithm can accommodate different postures, including those associated with walking, and has the advantage of being able to derive the desired results by setting usability features in an intuitive way.

Keywords: wearable robot; algorithm; rehabilitation; posture detection

check for updates

Citation: Lee, H.-W.; Lee, K.-O.; Chae, Y.-J.; Kim, S.-Y.; Park, Y.-Y. A Time-Scalable Posture Detection Algorithm for Paraplegic Patient Rehabilitation Using Exoskeleton-Type Wearable Robots. *Appl. Sci.* **2022**, *12*, 2374. https://doi.org/10.3390/app12052374

Academic Editor: Manuel Armada

Received: 27 December 2021
Accepted: 22 February 2022
Published: 24 February 2022

1. Introduction

With new developments in science and technology, social participation rates among disabled individuals have steadily increased. For example, paraplegic patients now have access to enhanced mobility platforms with advances in exoskeleton robot technology [1–3].

Safe and optimal operation of exoskeleton-type wearable robots by paraplegic patients requires that patients receive appropriate training. Here, we present the novel time-scalable posture detection algorithm (TSPDA) that can be used in exoskeleton training for paraplegic patients. TSPDA compares reference posture data (to be imitated by the patient) with posture data (recorded during training by the actual paraplegic patient) and extracts the evaluation target data required to measure the accuracy of the patient's recorded posture.

Some paraplegic patients may find it difficult to follow the target posture at the beginning of rehabilitation, while some may respond more quickly than expected. Thus, when comparing reference posture data with a paraplegic patient's posture data, the posture data of a paraplegic patient undergoing rehabilitation training may differ greatly from reference data in terms of movement speed. For example, reference posture data could indicate that it takes 5 s (s) to walk five steps. However, a paraplegic patient following the reference posture data may actually take 10 s to walk five steps. In this case, an accurate comparison between paraplegic patients and reference posture data are only possible when the time scale is doubled. It is necessary to find a way to determine whether the paraplegic patient has performed the correct motion, even if the time scale is different.

A posture comparison algorithm for the rehabilitation of paraplegic patients should flexibly adjust the time scale of reference data. The TSPDA introduced in this paper can overcome time-scale limitations; it can also allow comparison with other postures by changing the reference data.

Here, we use full body motion capture equipment featuring an IMU sensor [4] for postural analysis. In the past, IMU sensors were unreliable. However, the technology has greatly improved—reliability is now assured [5–7].

TSPDA compares the start and end frames of reference posture data with the complete data recorded from the paraplegic patient during training, and thereby avoid any time-scale constraints. The algorithm also utilizes the slice count method to reduce computational load: it limits the number of comparison samples between reference posture data and paraplegic patient data.

Considering that the posture accuracy of a paraplegic patient undergoing rehabilitation activities may be relatively low initially, the TSPDA has a function that allows for a minimum target accuracy to be set to accommodate the detection of posture data as it changes with the patient's range of motion.

The remainder of this paper is organized as follows. Section 2 describes related research. Section 3 presents the definitions of terms used in the algorithm and the algorithm's structure through diagrams and detailed module descriptions. Section 4 discusses an evaluation of the implemented algorithm and a performance analysis. Section 5 summarizes our findings and provides concluding remarks.

2. Related Work

A recent work [8] used a wearable robot with an exoskeleton to aid rehabilitation of paraplegic patients. The technology is advancing rapidly, but the learning curve is steep. Our TSPDA algorithm is aimed at flattening the curve and may have many applications in rehabilitation.

Traditional gait recognition algorithms only detect walking postures. The TSPDA directly compares complete raw data, which has been difficult for the algorithms proposed in several previous studies.

Existing gait recognition algorithms use a single sensor or several sensing units to extract the feature points of an actual physical posture as a sample, identifying the properties of each sample and applying them to actual data. These algorithms employ different approaches, depending on the sensor position (trunk, shank, foot, etc.), the data used (e.g., acceleration and/or angular velocity), and the number of sensors.

In this study, we used the Perception Neuron Full-body Tracking Sensor Kit (Noitom Ltd., Beijing, China) for posture detection; each sensor has a sampling rate of 120 Hz. Thus, we extracted reference posture data using more sensors and raw data than previous studies.

The types of sensors used in this study and previous studies include inertial measurement unit (IMU) and gyroscope sensors [9–13], in which the walking posture is ana-lysed from estimates of the relative position of the sensor via three-axis acceleration data. Previous studies have demonstrated the ability to recognize gait postures only, such as "heel strike" and "toe off" states, using a small number of sensor data [14].

Some researchers have achieved better accuracy using a footswitch with an IMU sensor [15–19]. However, the footswitch may interfere with the subject's walking process [19]. In contrast, our proposed algorithm uses several IMU sensors, without a footswitch.

One study found that gait speed can affect the operation of the algorithm when a single or a small number of IMU and gyroscope sensors are used with conventional gait detection algorithms [20]. In our case, however, because our TSPDA uses a full-body tracking sensor kit with many sensors, the gait speed has little effect on the algorithm's operation via time-scale changes.

Notably, the ability to extract raw posture data is relatively new, given that full-body tracking sensor kits have only recently become available. Thus, to date few studies have addressed this functionality, in which the detection algorithm can accommodate various postures and time scales.

3. Timescale-Scalable Posture Detection Algorithm

The terms used in the TSPDA are listed in Table 1.

Table 1. The terms used in the TSPDA.

Terms	Description
Reference graph	A two-dimensional graph corresponding to motion capture data for standard postures used in the algorithm
Target graph	A two-dimensional graph corresponding to motion capture data generated from a patient with lower extremity paralysis
Target minimum accuracy	Targeted minimum accuracy value (between 0 and 100)
Slice count	The number of the sample when comparing candidate graphs extracted from the reference graph and the target graph
Reference start/end frames	A set of Y values corresponding to the first and last X values of the reference graph
Sum start/end frames	A set of values obtained by comparing the target graph and reference start/end frames to determine the accuracy (lower is better)
Start/end candidate frames	A set of values that meets a condition with an accuracy equal to or greater than the target minimum accuracy when comparing the target graph with reference start/end frames
Candidates	A set of values storing all frames composed by the beginning and end of the graph that can be produced from start candidate frames and end candidate frames, and its accuracy when compared with the reference frame.

The operation sequence of the modules composing the TSPDA is shown in Figure 1 below.

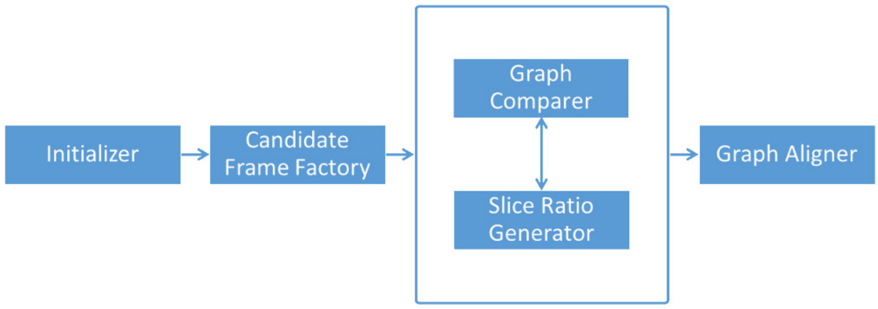

Figure 1. The operation sequence of the modules composing the TSPDA.

Each module is described in detail in the individual sub-sections that follow; the global variables are italicized in the descriptions.

3.1. TSPDA—Initialiser

The Initializer module defines the global variables (reference graph, target graph, slice count, target minimum accuracy, reference start frame, and reference end frame) required in the algorithm.

The Initializer module also compares the reference start frame and reference end frame with the target graph to calculate the relative accuracy (here, lower values are better). Additionally, it defines the sum start frames and sum end frames, which are accumulated values for each frame. The Initializer module is represented by the Algorithm 1 and Equations (1) and (2).

Algorithm 1: Pseudo code for the Initializer module of the TSPDA.

let (**referenceGraph**) be ([pre-recorded reference pose] composed with [X as time and Y as sensor data])
let (**targetGraph**) be ([recorded graph by patient] composed with [X as time and Y as sensor data])
let (**sliceCount**) be ([Desired Slice Count] when comparing the similarity between [[**targetGraph**] and the [candidates]])
let (**targetMinumumAccuracy**) be ([minimum target accuracy] between [**targetGraph** and **referenceGraph**])
let (**referenceStartFrames**) be ([Y values of **referenceGraph's** first X value])
let (**referenceEndFrames**) be ([Y values of **referenceGraph's** last X value])
let (**totalFrameCount**) be (Total count of X value for **targetGraph**)
let (**sumStartFrames**) be (empty list)
let (**sumEndFrames**) be (empty list)
for **x** = 1 to **totalFrameCount** do
let **sumStart** be 0
let **sumEnd** be 0
let **graph** = **targetGraph**[**x**]
let **targetMax** = max(**graph**)
let **targtMin** = min(**graph**)
let **targetDiff** = **targetMax-targetMin**
let **frameCount** = (frame count of the graph)
(equation #1)
(equation #2)
sumStartFrames[**x**] = **sumStart**
sumEndFrames[**x**] = **sumEnd**
end

$$sumStart = \sum_{y=1}^{frameCount} \left[\left| \frac{referenceStartFrames[y] - graph[y]}{targetDiff} \right| \right] \quad (1)$$

$$sumEnd = \sum_{y=1}^{frameCount} \left[\left| \frac{referenceEndFrames[y] - graph[y]}{targetDiff} \right| \right] \quad (2)$$

Equation (1) is a formula for calculating *sum start* (in which a lower number corresponds to higher accuracy), based on the reference start frames and the target graph.

Equation (2) is a formula for calculating the sum end by applying Equation (1) above to the reference end frames.

3.2. TSPDA—Candidate Frame Factory

The Candidate Frame Factory module calculates similarity by comparing the relative values of the sum start frames and the sum end frames calculated through the Initialiser module with the target frames.

The Candidate Frame Factory module is a module that defines start candidate frames and end candidate frames containing values having a similarity greater than or equal to the target accuracy.

The Candidate Frame Factory module is represented by the Algorithm 2 below.

Algorithm 2: Pseudo code for the Candidate Frame Factory module of the TSPDA.

```
let (diffSumStart) be max(sumStartFrames)-min(similarityStart)
let (diffSumEnd) be max(similarityStart)-min (similarityStart)
for x = 1 to totalFrameCount do
let (start) be [sumStartFrames[x]-min(sumStartFrames)]
let (similarityStart) be [100-{(start/diffSumStart) × 100}]
let (end) be [sumEndFrames[x]-min(simEndFrames)]
if (similarityStart >= targetAccuracy) then
add (x) to (candidateStartFrames)
end
if (similarityEnd >= targetAccuracy) then
add (x) to (candidateEndFrames)
end
end
```

3.3. TSPDA—Graph Comparer, Slice Ratio Generator

The Graph Comparer module calculates an arbitrary graph area using the start candidate frame and end candidate frame calculated through the Candidate Frame Factory module.

After the comparison is performed through the slice ratio values, candidates are defined as a set having similar values (where lower values are more accurate) with candidate start and end frames.

The Graph Comparer module is represented by the Algorithm 3 below.

Algorithm 3: Pseudo code for the Graph Comparer module of the TSPDA.

```
let (candidates) be (new empty list)
for (startCandidateFrame) in (startCandidateFrames) do
  for (endCandidateFrame) in (endCandidateFrames) do
    if (endCandidateFrame) > (startCandidateFrame) then
        let (sum) be (accumulated accuracy value between a new graph composed by
        startCandidateFrame and endCandidateFrame and targetValue using
         Slice Ratio Generator Module)
        add (sum, startCandidateFrame, endCandidateFrame) to (candidates)
    end
  end
end
```

3.4. TSPDA—Graph Aligner

The Graph Aligner module aligns the candidates calculated through the Graph Comparer module in ascending order and iterates through such that the item is added to the final result.

Additionally, when adding the item, the Graph Aligner module checks whether the item frame area of the already recorded item overlaps the area of the frame area to be newly registered. If the area overlaps, it is excluded from the results.

Candidates are sorted by ascending order by accuracy, which means an item recorded earlier has a higher accuracy. Thus, checking a previous registered item's area before adding an item to the result enables extraction of the item with the highest accuracy among the items that overlap the area.

The Graph Aligner module is represented by the Algorithm 4 below.

Algorithm 4: Pseudo code for the Graph Aligner module of the TSPDA.

order (**candidates**) by (**sum**) in (ascending order)
let (**results**) be (new empty list)
for (**startX**, **endX**) in (**candidates**) do
    let (**isValid**) be (true)
    for (**recordedStartX**, **recordedEndX**) in (results) do
        if (**recordedStartX** <= **endX** and **startX** <= **recordedEndX**) or (**startX** <= **recordedEndX** and **recordedStartX** <= **endX**) then
            set (**isValid**) to (false)
            break the loop
        end
    end
    if (**isValid**) is (true) then
        add (**start**, **end**) to (**results**)
    end
end

4. Experimental Results

The required personal computer specifications and environment used to drive the TSPDA are given Table 2 below.

Table 2. The required personal computer specifications and environment used to drive the TSPDA.

CPU	RAM	OS	Language	Compiler
i7-1165G7@2.8GHz	16 Gb	Windows11	C# (.NET 6.0)	MS Build

Figures 2 and 3 present the TSPDA results.

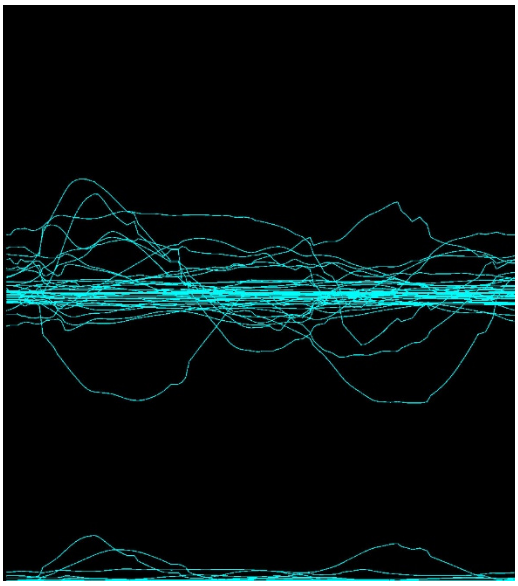

Figure 2. Reference graph.

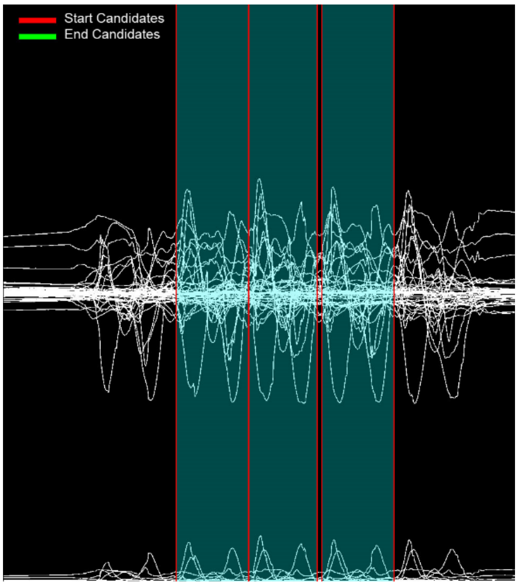

Figure 3. Target graph.

Figure 2 shows the posture of an ordinary person walking two steps, and Figure 3 presents data of a user imitating the posture of Figure 2 three times. Visually, it can be seen that the detected graph and the reference graph are different in length; however, the waveform itself is very similar.

Table 3 lists the data used to derive the results of the two pictures. All areas with the same properties as the reference graph were derived from the recorded posture data using the TSPDA.

Table 3. The data used to derive the results of the two pictures.

Data	Value
Target accuracy	65
Slice Count	10
Reference frame size	140
Execution time	54.89 ms (milliseconds)

Figures 4–8 show the TSPDA results when the slice count was nil (the optimal case) and 10, 25, 50, and 100. The results were similar.

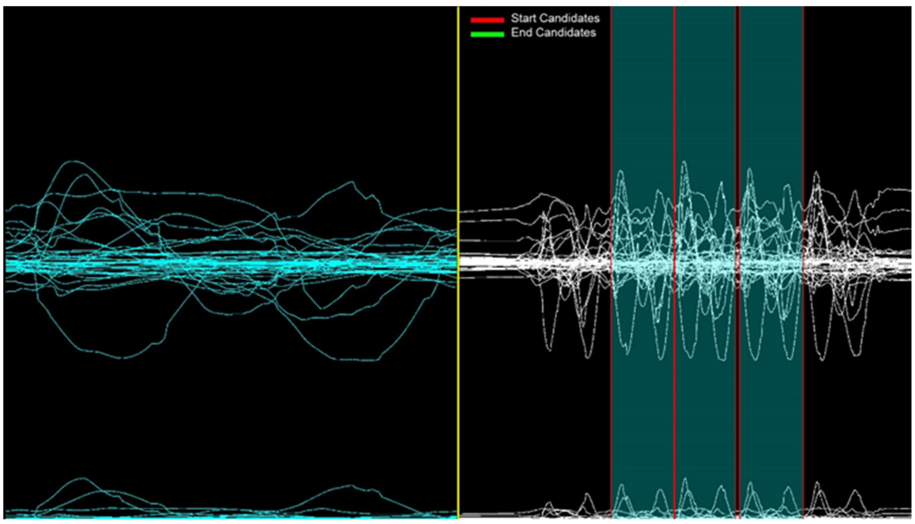

Figure 4. The result in the absence of slice counting.

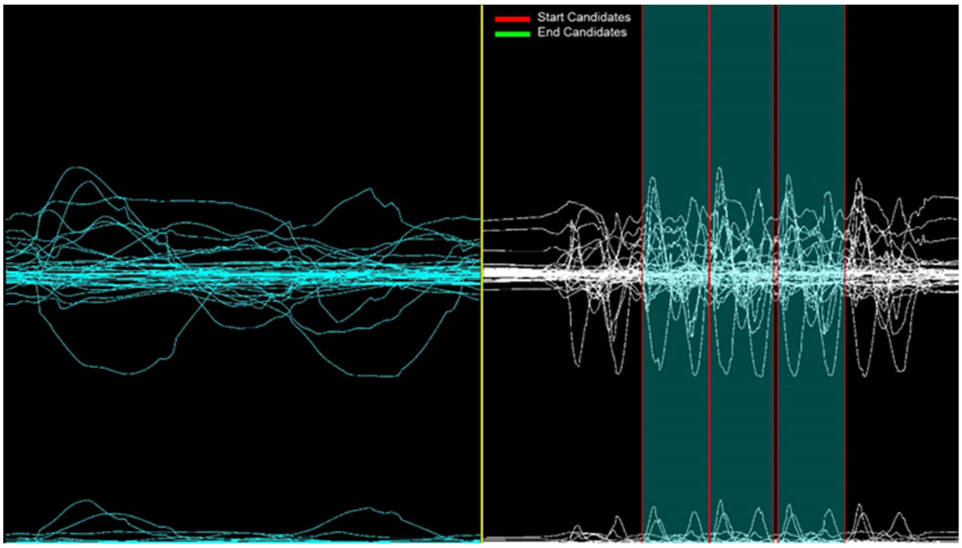

Figure 5. The result with a slice count of 10.

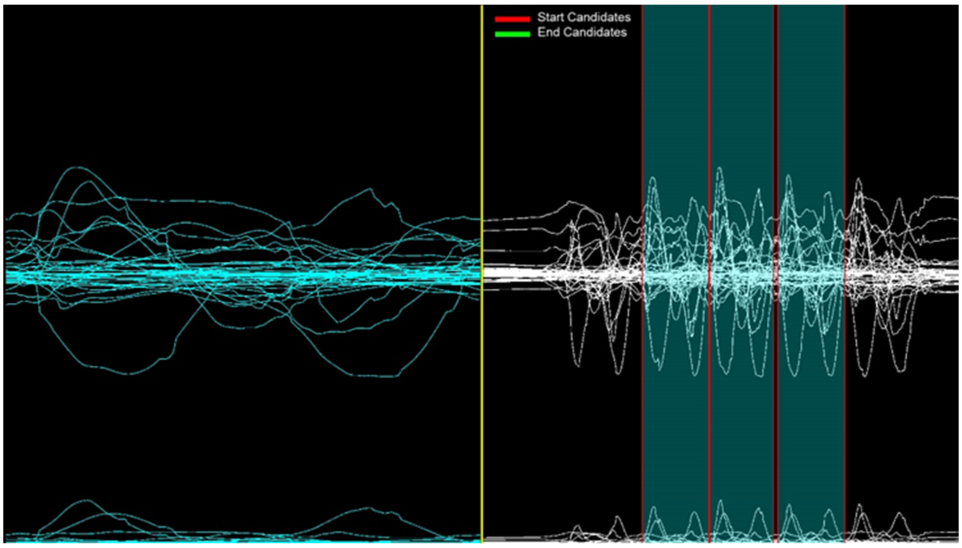

Figure 6. The result with a slice count of 25.

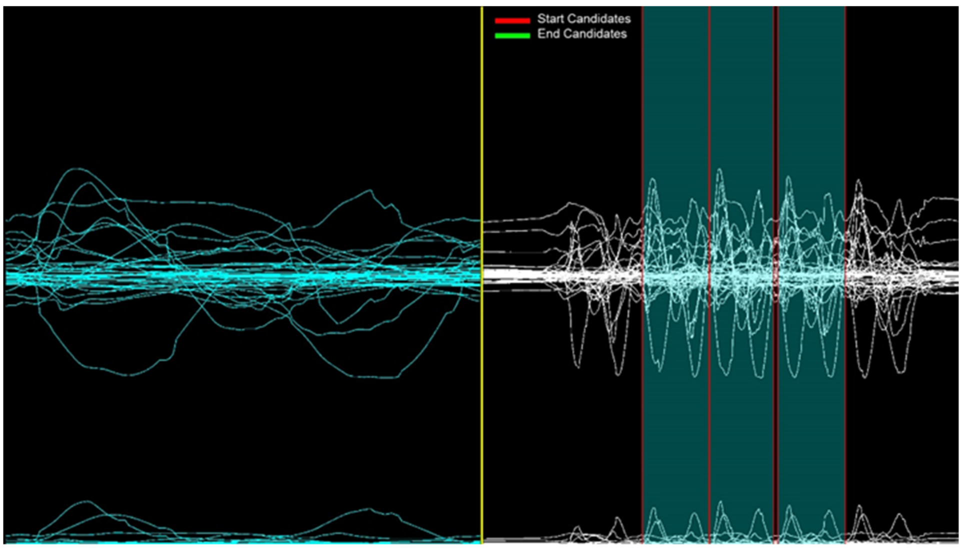

Figure 7. The result with a slice count of 50.

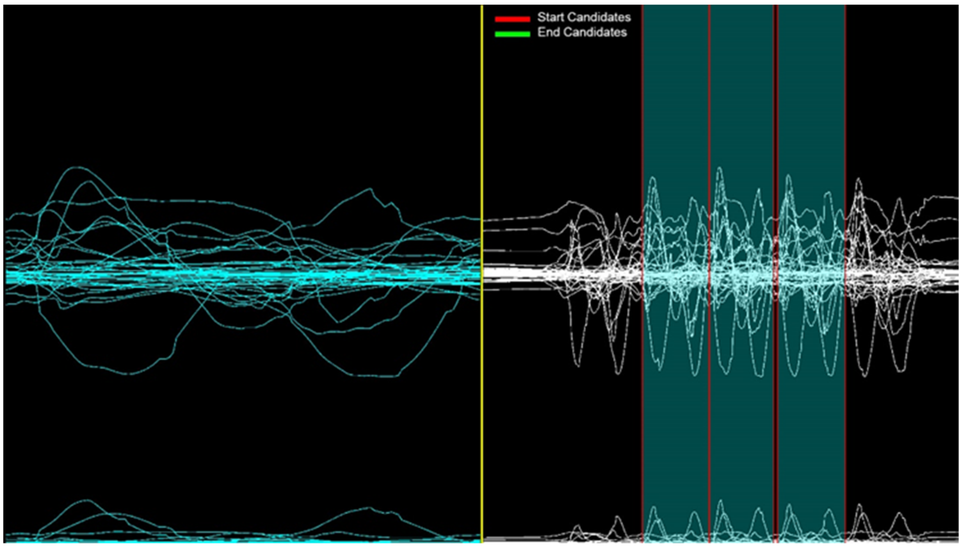

Figure 8. The result with a slice count of 100.

Figures 9 and 10 show that slice counting afforded accurate results (both visually and numerically) compared to the use of reference posture data only. Specifically, at a slice count of 10, execution was 7.67-fold faster than when slice counting was absent (the optimal case), but the accuracy was 96.5%. When the slice count was 25, execution was 6.24-fold faster and the accuracy 97.2%. When the slice count was 50, execution was 3.5-fold faster and the accuracy 98.7%. When the slice count was 100, execution was 1.9-fold faster and the accuracy 99.8%.

When slice counting is absent (the optimal case) the accuracy is 100% but execution is slow. The TSPDA slice count method thus facilitates individualized balancing of execution time and accuracy.

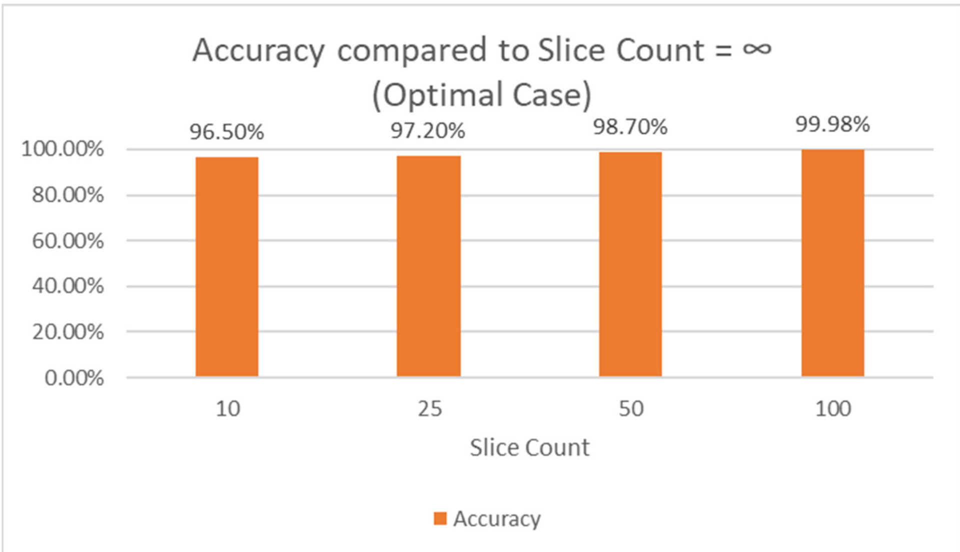

Figure 9. Accuracy compared to result in the absence of slice counting.

Figure 10. Execution time of the TSPDA by Slice Count.

5. Conclusions

This paper has introduced an algorithm that can efficiently compare how accurately a patient wearing an exoskeleton-type device performs a set of reference postures during rehabilitation training.

When the slice count method was applied, the accuracy did not differ significantly from the optimal value. However, application of the slice count method greatly enhanced the execution speed. Additionally, with the proposed TSPDA algorithm, the analysis target is not limited to the walking posture in posture analyses; the algorithm can universally accommodate various input postures using a full-body motion capture device.

Mobility solutions for people with physical disabilities continue to advance with new developments in robot technology, and high-performance assistance algorithms are needed to optimize patient rehabilitation and safety. The proposed TSPDA can be used to evaluate the training scale in the rehabilitation process necessary for the social advancement of paraplegic patients who cannot move their lower extremities. Given the broad scope of full-body motion tracking and the price declines in sensor components, we expect more availability and more demand for these devices as existing barriers are overcome. The TSPDA presented in this paper is expected to benefit society.

The TSPDA directly compares the raw data of motion capture equipment, thereby solving the problems associated with the limited postural analyses of the studies cited in Section 2; the TSDPA quickly and accurately defines data-rich areas using the real-time measures described in Section 4. TSPDA accuracy is assured by the fact that the initial settings can be modified. It is possible to obtain the desired results (via adjustment) even if the calculations are inaccurate.

Author Contributions: Conceptualization, H.-W.L., K.-O.L. and Y.-J.C.; Formal analysis, H.-W.L., K.-O.L., S.-Y.K. and Y.-Y.P.; Investigation, H.-W.L., K.-O.L., S.-Y.K. and Y.-Y.P.; Project administration, H.-W.L.; Resources, Y.-J.C.; Software, H.-W.L.; Supervision, K.-O.L., Y.-J.C. and Y.-Y.P.; Validation, H.-W.L., K.-O.L. and Y.-Y.P.; Writing—Original draft, H.-W.L.; Writing—Review and editing, H.-W.L., K.-O.L. and Y.-Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Sun Moon University Research Grant of 2021.

Institutional Review Board Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shi, D.; Zhang, W.; Zhang, W.; Ding, X. A review on lower limb rehabilitation exoskeleton robots. *Chin. J. Mech. Eng.* **2019**, *32*, 1–11. [CrossRef]
2. Jiang, J.G.; Ma, X.F.; Huo, B.; Zhang, Y.D.; Yu, X.Y. Recent advances on lower limb exoskeleton rehabilitation robot. *Recent Pat. Eng.* **2017**, *11*, 194–207. [CrossRef]
3. Seel, T.; Raisch, J.; Schauer, T. IMU-based joint angle measurement for gait analysis. *Sensors* **2014**, *14*, 6891–6909. [CrossRef] [PubMed]
4. Sers, R.; Forrester, S.; Moss, E.; Ward, S.; Ma, J.; Zecca, M. Validity of the Perception Neuron inertial motion capture system for upper body motion analysis. *Measurement* **2020**, *149*, 107024. [CrossRef]
5. Waqar, A.; Ahmad, I.; Habibi, D.; Hart, N.; Phung, Q.V. Enhancing athlete tracking using data fusion in wearable technologies. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–13. [CrossRef]
6. Germanotta, M.; Mileti, I.; Conforti, I.; Del Prete, Z.; Aprile, I.; Palermo, E. Estimation of human center of mass position through the inertial sensors-based methods in postural tasks: An accuracy evaluation. *Sensors* **2021**, *21*, 601. [CrossRef] [PubMed]
7. Cardarelli, S.; di Florio, P.; Mengarelli, A.; Tigrini, A.; Fioretti, S.; Verdini, F. Magnetometer-free sensor fusion applied to pedestrian tracking: A feasibility study. In Proceedings of the 2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT), Ancona, Italy, 19–21 June 2019; pp. 238–242.
8. Choi, H.; Lee, J.; Kong, K. A human-robot interface system for walkon suit: A powered exoskeleton for complete paraplegics. In Proceedings of the IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 21–23 October 2018; pp. 5057–5061.
9. Liu, J.; Zhang, Y.; Wang, J.; Chen, W. Adaptive sliding mode control for a lower-limb exoskeleton rehabilitation robot. In Proceedings of the 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), Wuhan, China, 31 May–2 June 2018; pp. 1481–1486.
10. Chen, M.; Huang, B.; Xu, Y. Intelligent shoes for abnormal gait detection. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 2019–2024.
11. Kardos, S.; Balog, P.; Slosarcik, S. Gait dynamics sensing using IMU sensor array system. *Adv. Electr. Electron. Eng.* **2017**, *15*, 71–76. [CrossRef]
12. Gujarathi, T.; Bhole, K. Gait analysis using imu sensor. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; pp. 1–5.
13. Hundza, S.R.; Hook, W.R.; Harris, C.R.; Mahajan, S.V.; Leslie, P.A.; Spani, C.A.; Spalteholz, L.G.; Birch, B.J.; Commandeur, D.T.; Livingston, N.J. Accurate and reliable gait cycle detection in Parkinson's disease. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2013**, *22*, 127–137. [CrossRef] [PubMed]
14. Panebianco, G.P.; Bisi, M.C.; Stagni, R.; Fantozzi, S. Analysis of the performance of 17 algorithms from a systematic review: Influence of sensor position, analysed variable and computational approach in gait timing estimation from IMU measurements. *Gait Posture* **2018**, *66*, 76–82. [CrossRef] [PubMed]
15. Jasiewicz, J.M.; Allum, J.H.; Middleton, J.W.; Barriskill, A.; Condie, P.; Purcell, B.; Li, R.C. Gait event detection using linear accelerometers or angular velocity transducers in able-bodied and spinal-cord injured individuals. *Gait Posture* **2006**, *24*, 502–509. [CrossRef] [PubMed]
16. Aminian, K.; Najafi, B.; Büla, C.; Leyvraz, P.F.; Robert, P. Spatio-temporal parameters of gait measured by an ambulatory system using miniature gyroscopes. *J. Biomech.* **2002**, *35*, 689–699. [CrossRef]
17. Lee, J.A.; Cho, S.H.; Lee, Y.J.; Yang, H.K.; Lee, J.W. Portable activity monitoring system for temporal parameters of gait cycles. *J. Med. Syst.* **2010**, *34*, 959–966. [CrossRef] [PubMed]
18. Salarian, A.; Russmann, H.; Vingerhoets, F.J.; Dehollain, C.; Blanc, Y.; Burkhard, P.R.; Aminian, K. Gait assessment in Parkinson's disease: Toward an ambulatory system for long-term monitoring. *IEEE Trans. Biomed. Eng.* **2004**, *51*, 1434–1443. [CrossRef] [PubMed]
19. Wentink, E.C.; Schut, V.G.; Prinsen, E.C.; Rietman, J.S.; Veltink, P.H. Detection of the onset of gait initiation using kinematic sensors and EMG in transfemoral amputees. *Gait Posture* **2014**, *39*, 391–396. [CrossRef] [PubMed]
20. Catalfamo, P.; Ghoussayni, S.; Ewins, D. Gait event detection on level ground and incline walking using a rate gyroscope. *Sensors* **2010**, *10*, 5683–5702. [CrossRef] [PubMed]