RATE

Journal of Rehabilitation and Assistive Technologies Engineering
Volume 9: 1–14
© The Author(s) 2022
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/20556683221117085
journals.sagepub.com/home/jrt
SAGE

A real-time algorithm for the detection of compensatory movements during reaching

Edward Averell, Don Knox and Frederike van Wijck

Abstract

Introduction: Interactive game systems can motivate stroke survivors to engage with their rehabilitation exercises. However, it is crucial that systems are in place to detect if exercises are performed correctly as stroke survivors often perform compensatory movements which can be detrimental to recovery. Very few game systems integrate motion tracking algorithms to monitor performance and detect such movements. This paper describes the development of algorithms which monitor for compensatory movements during upper limb reaching movements in real-time and provides quantitative metrics for health professionals to monitor performance and progress over time.

Methods: A real-time algorithm was developed to analyse reaching motions in real-time through a low-cost depth camera. The algorithm segments cyclical reaching motions into component parts, including compensatory movement, and provides a graphical representation of task performance. Healthy participants (*n* = 10) performed reaching motions facing the camera. The real-time accuracy of the algorithm was assessed by comparing offline analysis to real-time collection of data.

Results: The algorithm's ability to segment cyclical reaching motions and detect the component parts in real-time was assessed. Results show that movement types can be detected in real time with accuracy, showing a maximum error of 1.71%.

Conclusions: Using the methods outlined, the real-time detection and quantification of compensatory movements is feasible for integration within home-based, repetitive task practice game systems for people with stroke.

Keywords

Motion/ Posture Analysis, Motion Analysis Systems, Virtual Reality, Stroke Rehabilitation, Assistive Technology

Date received: 23 April 2021; accepted: 15 July 2022

Background

This paper describes upper limb motion tracking algorithms which monitor cyclical reaching motions in real-time, segmenting these complex movements into their component parts and detecting compensatory movements. The algorithm is embedded within a rhythm and music-based game system for stroke rehabilitation,[1,2] aiming to provide high dosage repetitive task practice while monitoring task performance to optimise therapeutic benefit.

Reduced upper limb functionality is common following stroke, with problems such as paresis, pain and spasticity being experienced throughout the hand, arm, and shoulder.[3] These impairments can inhibit a stroke survivor's ability to perform many activities of daily living, which can dramatically affect mood and quality of life.[4] Research indicates a high dosage of repetitive task practice is desirable for effective rehabilitation,[5,6] driving neuroplastic changes within the brain which underpin recovery.[7] Contemporary healthcare policy encourages early supported discharge from hospital where appropriate,[8] resulting in much of the

Glasgow Caledonian University, Glasgow, UK

Corresponding author:
Edward Averell, Glasgow Caledonian University, Cowcaddens Road, Glasgow G4 0BA, UK.
Email: edward.averell@gcu.ac.uk

Creative Commons Non Commercial CC BY-NC: This article is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 License (https://creativecommons.org/licenses/by-nc/4.0/) which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).

rehabilitation programme being conducted at home without expert supervision. The lack of constant supervision requires motivation and self-management from the stroke survivor to engage with the prescribed motions. However, stroke patients commonly experience depression,[9] and a lack of professional support to boost confidence means that some may find it difficult to undertake the prescribed exercises. These aspects are compounded by the COVID-19 pandemic, which has led to many stroke survivors missing out on crucial supervised rehabilitation.[10] There is therefore an urgent need to develop engaging interventions which the stroke survivor can perform safely and effectively at home.

Game technology has potential to offer assistance in this area and has been shown to have a positive impact on motivation and engagement with upper limb rehabilitation.[11] Games can be effective for motor recovery as the high dosage repetitive task practice necessary to facilitate motor learning can be embedded within an interactive entertainment platform, keeping the user engaged for prolonged periods of time and relaying crucial performance related feedback to the user and health professionals.[12] When used as an adjunct to conventional therapy, games have shown to be beneficial in motor recovery.[13] However, the design of the system is crucial,[14] with games which deliver established neurorehabilitation principles providing greater positive impact on motor function.[15]

Game systems can potentially motivate the user to engage with the prescribed rehabilitation exercises. However, it is crucial that systems are in place to detect if these exercises are performed correctly as stroke survivors often rely upon alternative movement strategies to compensate for limited functional ability.[16] When aiming to reach forward, these compensatory movements typically include increased trunk flexion[17,18] and rotation[19] While such movements can allow for the completion of a task, they can be detrimental to recovery as the original impairments (commonly reduced shoulder flexion and elbow extension) are not addressed.[20,21] Therefore, it is vital that robust motion tracking algorithms are embedded within game systems to monitor task performance, detecting compensatory motions which can hinder progress. However, motion tracking and analysis is rarely implemented within commercially available game systems with any degree of sophistication. In a review of game-based rehabilitation systems, Tamayo-Serrano et al.[22] state that from a total of 32 papers, only 18% of systems did not require the presence of a therapist to monitor patient progress. This raises the significant concern that the user may engage with the game system but may not perform the prescribed motions correctly, limiting the effectiveness of the intervention and potentially facilitating harmful activity.

Various methods of detecting compensatory movement have been employed throughout the literature. The generally accepted 'gold standard' for measuring and analysing human movement is through high-resolution cameras which observe markers placed at key positions on the human body. While these systems provide the most accurate results, they are unfeasible for home use as they are large, expensive, require expert setup knowledge, and are therefore typically confined to laboratory or clinical settings. Aprile et al.[19] utilised a marker-based system to measure reaching strategies of stroke survivors. From the displacement of markers placed at key anatomical landmarks, each reaching motion was segmented into distinct components - arm elongation, trunk forward inclination and trunk rotation. Each component was allocated a percentage contribution to reach, allowing for the level of compensation to be numerically quantified. However, the method proposed by Aprile et al.[19] groups all upper limb motion under 'arm elongation', therefore disregarding other sources of end effector displacement such as shoulder rotation. Additionally, the method does not offer real-time operation which is crucial for integration within game-based systems to provide motivational and performance feedback to the user.[12] However, it is acknowledged that all systems operate with a delay. Classification algorithms which aim to detect and categorise human motion have been explored.[23-28] However, these implementations do not typically quantify the amount of compensation, which is crucial for health professionals to evaluate progress over time. Hand-held game controllers have been investigated,[29] but these only capture the position of the controller device which some stroke survivors may find the difficult to grasp.[30] By contrast, low-cost depth cameras such as the Microsoft Kinect[31] circumvent these concerns by providing hands free interaction, capture a variety of anatomical positions, and omit the need for expert user knowledge. Implementations utilising depth cameras include monitoring of correct sitting posture,[32] assessment of joint angles to establish correct movement,[33] translation of movement to game input messages,[34] simple reaching games,[35] and recognition of actions and the plane of motion integrating a joint coordinate system.[36,37] However, these implementations generally consider single movements in isolation, use joint angles which are prone to error in consumer level cameras,[38] and do not detect and quantify the relationship between several compensatory actions in real-time. Such factors are essential for describing complex movements, setting rehabilitation goals, and conducting longitudinal analysis of performance. Yet low-cost commercially available systems are potentially affordable for routine implementation in healthcare systems.

To this end, this paper describes an algorithm which monitors reaching motions, recorded by a markerless consumer level motion capture camera[31] in real-time, detecting and quantifying the use of various compensation strategies. The algorithm advances previous work[19] by further segmenting upper limb motion to include shoulder rotation and providing real-time operation. The algorithm has been designed to operate within a repetitive task practice game system where reaching motions are performed to the rhythm of self-selected music, i.e. continuously on loop.[2] The algorithm observes the displacement of key joints, segments cyclical movements into discrete actions in real-time, and uses a series of calculations to establish the type of movement and level of compensation used to complete the task. A reaching motion has been categorised into four movement types – Forward Reach (FR), Trunk Forward inclination (TF), Trunk Rotation (TR) and Shoulder Rotation (SR), separating the movement into component parts and allocating each feature a percentage contribution to reach. This results in a single, dimensionless value for each movement type, which is subsequently used in simple graphical plots which aim to provide an easy to understand, simplified representation of task performance to inform stroke survivors and health professionals.

The next section of this paper presents the design of the algorithms in detail. Following this, the accuracy of the algorithms performance in real-time was tested by recruiting ten able-bodied study participants to trial the system. Finally, the potential benefits of the system's design and intended applications are discussed.

System development

Input and development environment

A real-time algorithm has been developed which takes input from the Microsoft Kinect V2 sensor.[31] As an input device, Kinect V2 has been shown to provide sufficient accuracy for clinical measurement of motor function.[39] Users interact with the system by facing a single Kinect camera at an approximate distance of 1.5 m and 70 cm from the floor (Figure 1).

The Kinect data is collected at a sampling rate of 30 Hz. Joint positions are inferred within the Neo-Kinect plugin via the Unreal Engine game development environment.[40] Within the game system, the primary task is to perform reaching motions between two or more onscreen target points in synchronous with an audible rhythm provided by user selected music.[2] The positions of the wrist, elbow, shoulder, and central trunk (termed spine) are collected and processed (Figure 2). Both the left and right upper limb are available to use in game.

Algorithm overview

The algorithm monitors the positions of key joints to describe task performance when reaching. The three-dimensional location of each joint is compressed and projected onto the sagittal place to observe displacement in an anterior direction. The algorithm segments a reaching motion into four movement types:

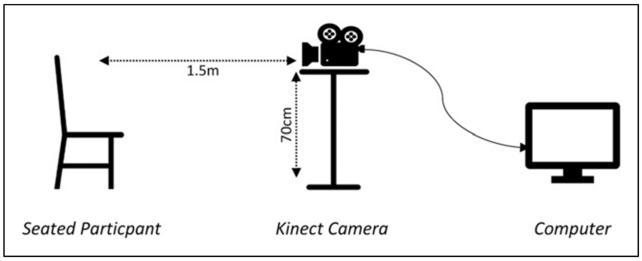

Figure 1. System setup.

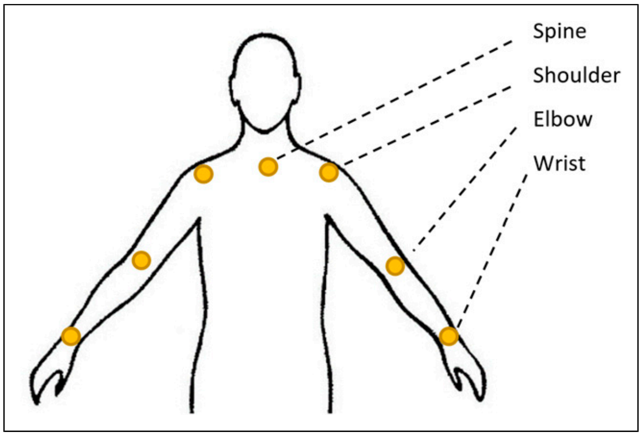

Figure 2. Joints captured via Kinect V2 camera. Both the left and right upper limb are available to use in game.

- Trunk Forward (TF) – Trunk flexion within the sagittal plane
- Trunk Rotation (TR) – Axial Rotation of the trunk on the transverse plane
- Shoulder Rotation (SR) – Internal/external shoulder rotation on the transverse plane
- Forward Reach (FR) – Shoulder flexion and elbow extension on the sagittal plane

By considering only the positions of the hand and trunk, the combination of 'forward reach' (elbow extension and shoulder flexion in the sagittal plane) and trunk compensation (leaning with the trunk in an anterior direction) could describe task performance when reaching forward. For example, if a user were to keep their trunk stationary and use only the upper limb to reach a target point with their hand, forward reach will account for 100% of the movement. Conversely, if the user does not extend their arm and reaches the desired position solely by leaning their trunk forward, trunk compensation will account for 100% of the movement. These measurements can be obtained by observing the level of point-to-point *displacement* in an anterior direction.

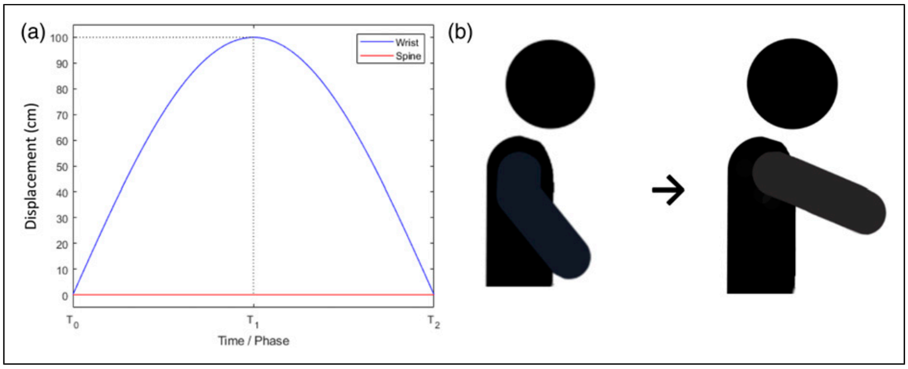

Figure 3. (a) – Simulated displacement with no trunk compensation and; (b) - illustration of the movement where the trunk is stationary and the upper limb reaches forward to complete the motion. The *Y* axis represents displacement in an anterior direction in centimetres (100 signifying maximum reach) and the *X* axis represents the phase of the motion.

A simulation of displacement of the hand and trunk through a reaching motion on a single axis over time is modelled below. The *X* axis represents time, and the *Y* axis displays displacement in an anterior direction in centimetres. Figure 3(a) (b) shows displacement using only forward reach and no trunk compensation, whereas Figure 4(a) (b) shows the same motion, but with trunk compensation.

In the above example, the task of reaching can be separated into key phases – the forward motion to reach the target (T_0-T_1), the instant the hand reaches the target (T_1), and the reversal of the motion and return to the starting position (T_1-T_2). Figure 3 and Figure 4 illustrate displacement considering only the position of the hand and trunk. However, to provide a more comprehensive analysis of movement and give insight into features such as rotation, the position of the elbow and shoulder must also be considered. The simulation shown in Figures 2 and 3 show the starting position of each joint at an arbitrary position. However, as each joint occupies a unique anatomical position, it is necessary to observe the starting position for each joint (the position of each joint at T_0) and observe its deviation from this position to T_1. Figure 5 models a movement arc for the wrist, elbow, shoulder, and spine in the sagittal plane with varying displacement levels for each joint. These values are highlighted at the beginning (T_0) and peak (T_1) of the reach cycle.

By taking the *Y* axis values at T_0 and T_1 shown in Figure 4 and calculating the difference, the total *Y* axis displacement of the wrist, elbow, shoulder, and spine is obtained (Table 1).

The combination of these four displacement values can be used to calculate contribution in the sagittal plane to the task of reaching forward with reference to Trunk Forward inclination (TF), Trunk Rotation (TR), Shoulder Rotation (SR) and Forward Reach (FR). The key measurement is displacement of the end effector in the sagittal plane, which is the wrist in the above example. All features are calculated relative to this motion, normalising all values to a scale of 0–1.

Trunk forward inclination

Trunk Forward inclination (TF) is measured by assessing the relative levels of displacement between the spine and wrist in an anterior direction This calculation does not consider the positions of the shoulder and elbow. To calculate this value, the total spine displacement (ΔSp) is divided by the total wrist displacement (ΔWr), then multiplied by 100. This results in a value which details the percentage Trunk Forward inclination when reaching (Equation (1)).

Equation (1) - Trunk Forward Inclination

$$TF(\%) = 100 * \left(\frac{\Delta Sp}{\Delta Wr}\right)$$

Using the displacement values shown in Table 1 results in the following calculation (Equation (2)), revealing that 38.89% of the forward reaching motion shown in Figure 4 was achieved through forward leaning of the trunk.

Equation (2) - Trunk Forward Inclination worked example

$$TF(\%) = \ 100 * \left(\frac{14}{36}\right)$$

$$\boldsymbol{TF = 38.89\%}$$

Trunk rotation

Trunk Rotation (TR) describes the amount of trunk axial rotation in the transverse plane performed when reaching. To obtain this measurement, the movement is not measured on the transverse plane, but projected onto the sagittal plane and the relative positions of key joints observed. As greater displacement in an anterior direction seen in the shoulder than the spine reveals a rotation of the trunk (assuming there is no

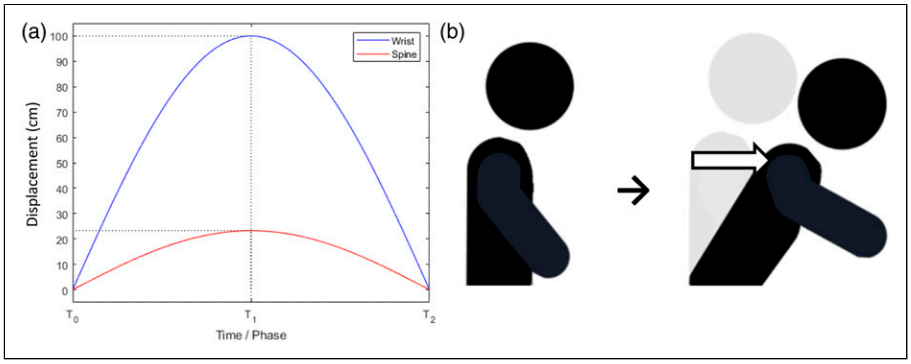

Figure 4. (a) – Simulated displacement *with* trunk compensation and; (b) - illustration of the movement where the trunk contributes to the motion through forward leaning. As in Figure 2, the *Y* axis represents displacement in an anterior direction in centimetres (100 signifying maximum reach) and the *X* axis represents the phase of the motion.

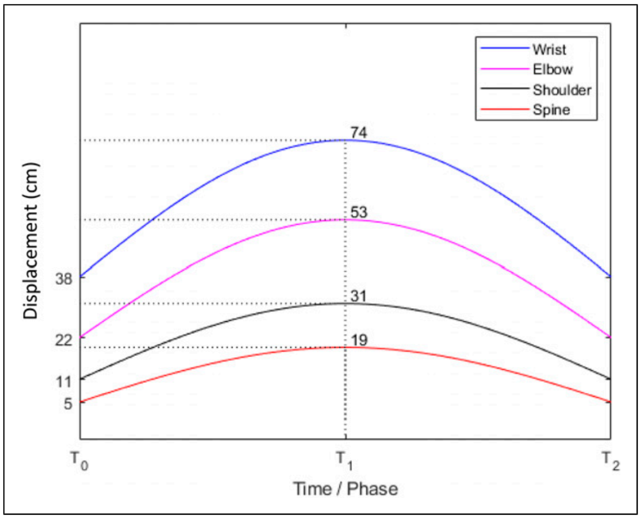

Figure 5. Displacement of joints from non-zero start point in the sagittal plane.

Table 1. Joint positions at T_0 and T_1 and total *Y* axis displacement in arbitrary units.

Joint	T_0 *Y*-Axis position	T_1 *Y*-Axis position	*Y* Axis displacement (T_1 - T_0)
Wrist	38	74	36
Elbow	22	53	31
Shoulder	11	31	20
Spine	5	19	14

shoulder protraction), the difference between shoulder displacement (ΔSh) and spine displacement (ΔSp) relative to total wrist displacement (ΔWr) in the sagittal plane, details the contribution of trunk rotation. Therefore, the initial calculation establishes the difference between shoulder and spine displacements, taking the absolute value to ensure positive output. Once established, this value is divided by the total wrist displacement and multiplied by 100 to establish the percentage contribution of Trunk Rotation (Equation (3)).

Equation (3) - Trunk Rotation

$$TA(\%) = 100 * \left(\frac{(|\Delta Sh - \ \Delta Sp|)}{\Delta Wr}\right)$$

Using the displacement values shown in Table 1 results in the following calculation (Equation (4)), revealing that 16.67% of the forward reaching motion was achieved through rotation of the trunk.

Equation (4) - Trunk Rotation worked example

$$TA(\%) = 100 * \left(\frac{6}{36}\right)$$

$$\boldsymbol{TA = 16.67\%}$$

Upper limb contribution

Describing the contribution of the upper limb to reaching brings greater complexity due to the shoulder. As the position of the hand can be changed through internal and external rotation of the shoulder, shoulder rotation must be factored into any calculations. By observing the shoulder, elbow and wrist's relative displacement in an anterior direction, the contribution of the upper limb can be segmented into two key movements:

- *Shoulder Rotation* - Internal and external shoulder rotation in the transverse plane
- *Forward Reach* - Shoulder flexion and elbow extension in the sagittal plane

In summary, displacement in an anterior direction of both the elbow and wrist indicates *Forward Reach*, whereas displacement in an anterior direction of the wrist with less or

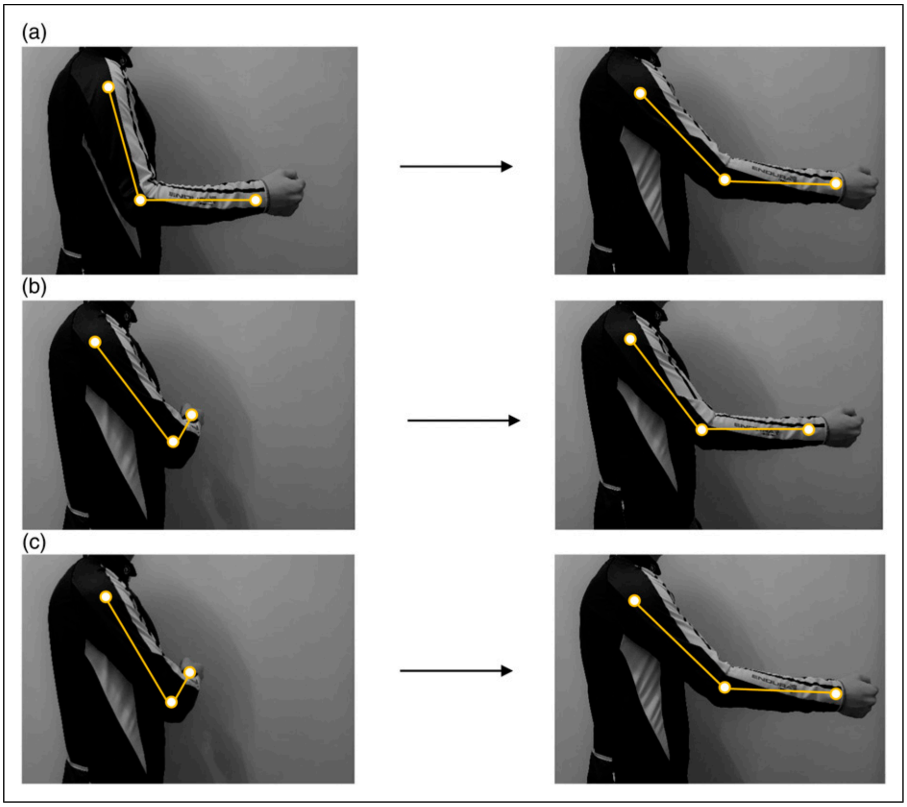

Figure 6. (a) – Wrist displacement by forward reach only. (b) - Wrist displacement by shoulder rotation only. (c) – Wrist displacement by a combination of shoulder rotation and forward reach.

no displacement of the elbow indicates shoulder rotation. Figure 6 shows three scenarios of wrist displacement in an anterior direction - forward reach only (Figure 6(a)), shoulder rotation only (Figure 6(b)), and a combination of both forward reach and shoulder rotation (Figure 6(c)).

Shoulder rotation

Shoulder Rotation (SR) describes internal and external shoulder rotation in the transverse plane, and observes the positions of the wrist, elbow, and shoulder for all calculations. As the wrist and elbow are connected through a rigid body which links their position (the forearm), it is appropriate to conclude that if displacement of the wrist in an anterior direction is observed with less or no displacement of the elbow, shoulder rotation has occurred. Therefore, to assess what displacement can be attributed to shoulder rotation, it is necessary to 'remove' any displacement of the wrist caused by displacement of the elbow.

Subtracting the difference between shoulder (ΔSh) and elbow (ΔEl) displacement from the difference between shoulder and wrist (ΔWr) displacement removes any displacement of the wrist caused by displacement of the elbow. Once displacement of the wrist caused by displacement of the elbow is removed, it can be concluded that any remaining wrist displacement is caused by shoulder rotation and not forward reach. The absolute value of each calculation is taken, and the total multiplied by 100 to obtain the percentage contribution of shoulder rotation (Equation (5)).

Equation (5) - Shoulder Rotation

$$SR = 100 * \left(\frac{(|\Delta Wr - \ \Delta Sh|) - (|\Delta Sh - \ \Delta El|)}{\Delta Wr} \right)$$

Using the displacement values shown in Table 1 results in the following calculation (Equation (6)), revealing that 13.88% of the reaching motion was achieved through rotation of the shoulder.

Equation (6) - Shoulder Rotation worked example

$$SR = 100 * \left(\frac{(|36 - 20|) - (|20 - 31|)}{36} \right)$$

$$SR = 100 * \left(\frac{16 - 11}{36} \right)$$

$$\boldsymbol{SR = 13.88\%}$$

Forward reach

Forward Reach (FR) is a combination of shoulder flexion and elbow extension in the sagittal plane, and observes the positions of the wrist, elbow, and shoulder for all calculations. As with shoulder rotation, the position of the elbow is critical in determining the level of forward reach that has occurred. In summary, if the wrist is displaced in an anterior direction through displacement of the elbow, it is concluded that forward reach has occurred.

As the difference between wrist (ΔWr) and elbow (ΔEl) displacement denotes the level of shoulder (ΔSh) rotation that has occurred, subtracting this value from the difference between wrist and shoulder displacements removes displacement of the wrist caused by shoulder rotation. Therefore, the remaining value is the level of wrist displacement caused by displacement of the elbow, revealing the level of forward reach. The absolute value of each calculation is taken, and the total multiplied by 100 to obtain the percentage contribution of forward reach (Equation (7)).

Equation (7) - Forward reach

$$FR = 100 * \left(\frac{((|\Delta Wr - \Delta Sh|) - (|\Delta Wr - \Delta El|))}{\Delta Wr} \right)$$

Using the displacement values shown in Table 1 results in the following calculation (Equation (8)), revealing that 30.56% of the reaching motion was achieved through forward reach.

Equation (8) - Forward reach worked example

$$FR = 100 * \left(\frac{(|36 - 20|) - (|36 - 31|)}{36} \right)$$

$$FR = 100 * \left(\frac{16 - 5}{36} \right)$$

$$\boldsymbol{FR = 30.56\%}$$

Data presentation

The previous calculations reveal that for the example shown in Figure 4, trunk forward inclination contributed 38.88%, trunk rotation contributed 16.67%, shoulder rotation contributed 13.89% and forward reach contributed 30.56%, totalling 100% (Table 2).

This information can be presented graphically, providing an easy to understand visual representation of user performance. Figure 7 shows the numerical information in Table 2 displayed in a stacked bar plot - Forward Reach (FR) is displayed in green, Shoulder Rotation (SR) in light blue, Trunk Rotation (TR) in dark blue, and Trunk Forward (TF) in red. Presenting the information in this manner negates the need for the therapist to interpret large amounts of numerical data. For example, if an increase in forward reach and a decrease in trunk forward inclination is set as a performance goal, it is desirable to see larger areas of green (FR) than red (TF) in any plots. By observing a series of reach cycles, or taking the mean of several cycles, changes in performance over time can be illustrated. For example, Figure 8 shows a series of reach cycles with minimal compensation, whereas Figure 9 shows a series of reaching motions with increased compensation.

Segmenting reach cycles in real-time

As the algorithm is embedded within a game which involves moving the hand between onscreen target point in synchronous with an audible rhythm,[1,2] accurate, real-time gathering of joint displacement values is essential. As the user is tasked with repeating the reaching motion on loop, they are placed in a *constant* state of assessment as they are continuously repeating the exercise. Therefore, it is crucial that cyclical motions can be accurately segmented into discrete actions in real-time, clearly defining the start and end points of the cycle with accuracy. Expanding upon the single reach cycle seen in Figure 4, a more representative data stream may appear as modelled in Figure 10. For simplicity, only the wrist position is displayed. However, while representative of a cyclical reaching motion, the movement of stroke survivors will generally be less smooth than presented.

As the system utilises a motion capture camera for user input, there is no physical contact point to signify the start and end points of the motion. Therefore, a method of defining the beginning and end of the reach cycle must be achieved. From within the game system, successful arrival at an onscreen target point could signify the start and end points of the movement cycle. However, each user will have differing functional ability, with the possibility of some users failing to reach any of the target points within the prescribed timeframe. While reaching the target point is a game play goal, the movement itself is of utmost importance and it is therefore crucial that attempts *towards* the goal are observed and analysed. Therefore, to capture all relevant movements and signify the start and end points of a cycle, a threshold level which signifies the user's arrival and return points is established. This threshold value is a percentage of the user's total reach. In summary, if the user's wrist crosses and returns to within a given percent of the minimum and maximum reach, an attempt at a reaching motion is signified and further analysis defines the start and end points of the cycle (Figure 11).

Using the concept outlined above, a real-time algorithm has been developed to segment movement data into reach cycles and collect key data required to calculate each movement type's contribution to reach. The algorithm observes the position of all joints during movement, performing a local minima and maxima search to gather their

Table 2. Percentage contributions to reach.

Movement type	Contribution, %
Trunk forward inclination	38.89
Trunk rotation	16.67
Shoulder rotation	13.88
Forward reach	30.56
Total	**100**

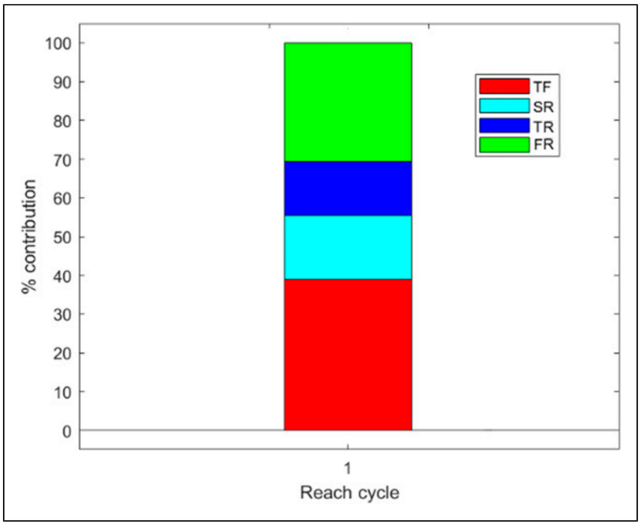

Figure 7. A single reach cycle showing Trunk Forward (TF), Shoulder Rotation (SR), Trunk Rotation (TR) and Forward Reach (FR).

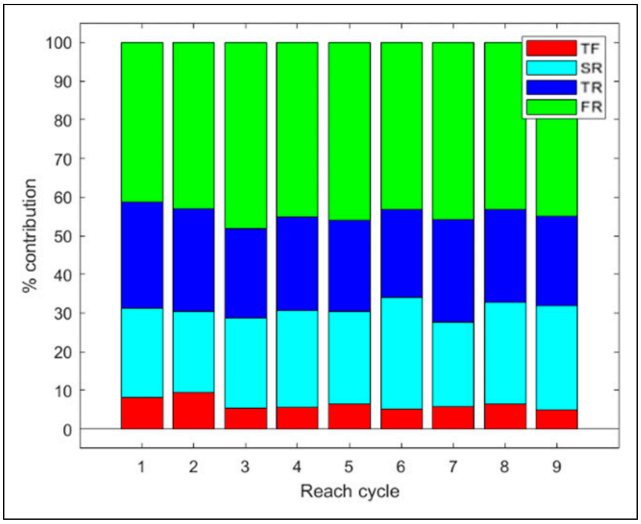

Figure 8. Minimal trunk compensation.

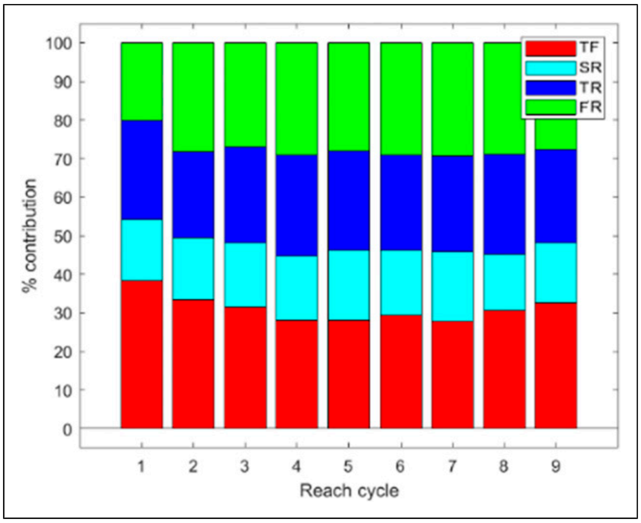

Figure 9. Increased trunk compensation.

position and time at the beginning and peak of each reach cycle.

The algorithm begins by establishing the user's maximum elbow flexion through measuring the maximum achievable distance between the wrist and ipsilateral shoulder. This value is used to establish thresholds as illustrated in Figure 10. The threshold level is established by obtaining a percentage of the user's maximum reach (Equation (9)).

Equation (9) - Threshold calculation

$$Threshold = MaxArmExtension * 0.x$$

Using the threshold value, minimum and maximum threshold 'zones' are set. A baseline position is then established for the wrist, elbow, shoulder, and spine through zeroing the coordinate position of each joint. No specific posture is required. However, to ensure consistency, the participant was asked to sit comfortably with their dominant hand on the closest target and this process was triggered by the researcher. These values are stored and used to calculate displacement in an anterior direction by measuring deviation from this point.

Through a series of conditional statements, the position of the wrist is monitored for entering and leaving the threshold zones in sequential order. Upon the wrist entering a threshold zone, the location of each joint (J) is stored in discrete arrays (denoted $J_{Max\ /}\ J_{Min}$) along with the time the value was captured (T_{Max}/T_{Min}). Once a complete reach cycle is detected, the values in all arrays are passed on to establish key data to detect and quantify compensatory movement (Algorithm 1).

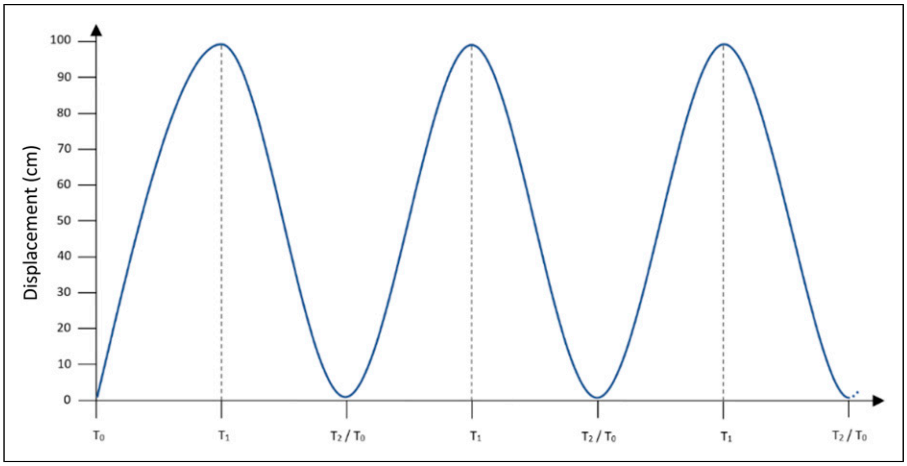

Figure 10. Modelled displacement of the wrist on the anteroposterior axis over multiple reach cycles.

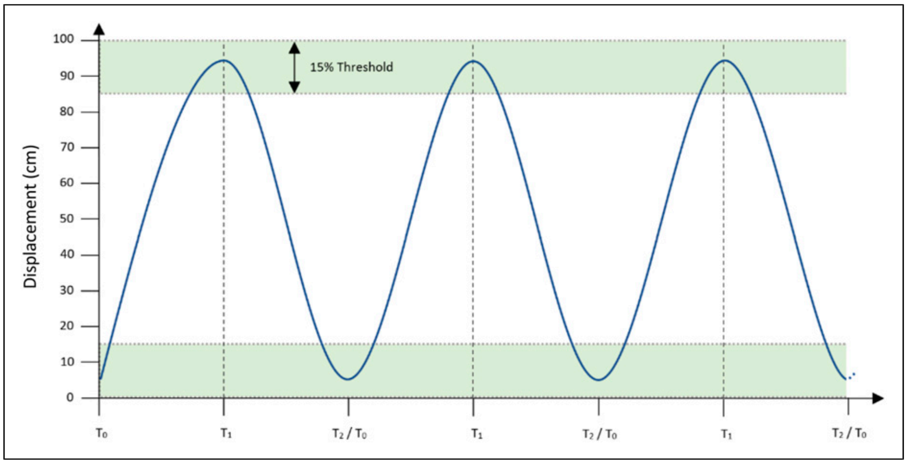

Figure 11. Modelled displacement of the wrist on the anteroposterior axis with threshold levels.

```
Input: MinThreshold, MaxThreshold, Joint (J) Positions
  » initialise threshold Booleans
  OverMax ← False
  UnderMin ← False
  » performed on each frame of game play
  for each frame
    » has the wrist crossed the max reach threshold?
    if WristPosition ≥ MaxThreshold
      OverMax ← True
      » while over the max threshold, store each
        joint's position and time in discrete arrays
      Store all joint positions and time [$J_{Max}$, $T_{Max}$]
    end if
    » has the wrist crossed the min threshold and
previously crossed the max threshold?
    if WristPosition ≤ MinThreshold &&
      OverMax = True
      UnderMin ← True
      » while under the min threshold:
      Store all joint positions and time [$J_{Min}$, $T_{Min}$]
    end if
    » Finally, when the wrist has advanced beyond
      the minimum threshold, reset Booleans
    if WristPosition ≥ MinThreshold &&
      UnderMin = True
      » pass on all joint position and time arrays
        for further processing
      Export joint position and time arrays [$J_{Max}$,
        $J_{Min}$, $T_{Max}$, $T_{Min}$]
      OverMax ← False
      UnderMin ← False
      end if
   end for
```

Algorithm 1 – Capture of minimum and maximum joint displacements and times

Wrist displacement

To obtain the position of wrist at the beginning of the reach cycle (WrT_0), the minimum value of the array gathered when the wrist was under the minimum threshold (WR_{Min}) is found. Similarly, to gather the position of the wrist at the peak of the reach cycle (WrT_1), the maximum value of the array gathered when the wrist was over the maximum reach threshold (WR_{Max}) is found.

There is potential for the minimum value of the array gathered at minimum extension (WrT_0) to be negative as the wrist may be retracted beyond the initialisation position. To ensure only positive values are used, the position of the wrist at the beginning of each reach cycle is checked for native values and the absolute value taken if found to be true. This value is stored (WR_{offset}) then added to the *maximum* displacement value (WrT_1). The *minimum* displacement value (WrT_0) is then set to zero, which maintains overall displacement and shifts data into a positive range (Algorithm 2).

```
WrT$_0$ ← min(WR$_{Min}$) » get minimum of array
   WrT$_1$ ← max(WR$_{Max}$) » get maximum of array
   if WrT$_0$ < 0
      WR$_{offset}$ ← |WrT$_0$|
      WrT$_1$ ← WrT$_1$ + WR$_{offset}$
      WrT$_0$ ← 0
   end if
```

Algorithm 2 - The position of the wrist at minimum (WrT_0) and maximum (WrT_1) of the reach cycle is found and displacement values are shifted into a positive range.

Proximal joint displacement

As the collection of positional information for the wrist, elbow, shoulder, and spine is triggered by the wrist entering and leaving the threshold zones, data for all joints are gathered simultaneously. Therefore, the array indices of the minimum and maximum wrist displacement are used to obtain the position of all proximal joints at these times. From this, elbow, shoulder, and spine displacements are gathered, and the data checked for negative values (Algorithm 3).

```
Input: Joint arrays [J$_{min}$, J$_{max}$], Wrist positions
[WrT$_0$, WrT$_1$]
   for each joint (J)
      » get displacement of joint at wrist T$_0$ & T$_1$
        indices
      JT$_0$ ← J$_{min}$[idx(WrT$_0$)]
      JT$_1$ ← J$_{max}$[idx(WrT$_1$)]
      if JT$_0$ < 0
         J$_{offset}$ ← |JT$_0$|
         JT$_1$ ← JT$_1$ + J$_{offset}$
         JT$_0$ ← 0
      end if
   end for
```

Algorithm 3 – Proximal joint displacement.
The positions of the wrist at minimum and maximum teach are used to locate the position of all proximal joints at the point in time.

From the minimum and maximum values gathered through algorithms 2 and 3, joint displacement is calculated, and compensatory motion is detected and quantified in relation to TF, TA, SR, and FR as previously outlined.

In game filtering

The raw data generated by Kinect contains a significant amount of noise. Therefore, real-time data smoothing within the game engine is essential to minimise potential for data used to segment reaching motions to be influenced by noisy sensor input. If a tracking error caused a signal spike in opposing directions for two significant joints, the relative displacement of (or, distance between) these joints would be recorded as greater than occurred in practice. This would cause the real-time local maxima and minima search to report incorrect displacement values, resulting in system error. Therefore, trajectory data is smoothed in real-time within the game engine to ensure all data gathered to calculate compensatory movement is as accurate as possible.

If sensor data is filtered offline (i.e. not in 'real-time'), aspects such as processing latency introduced by computationally expensive filtering algorithms are of no concern. However, for real-time data smoothing, a balance must be drawn between the method of smoothing employed and the latency incurred. The data is therefore processed using a moving average filter as it offers effective real-time smoothing, incurs minimal latency, and is computationally inexpensive. Equation (10) shows the calculation performed on Kinect input data on each frame of gameplay, where *out* is the smoothed output, *prev* is the previous smoothed output, *b* is the scaling factor (a value between 0 and 1), and *raw* is the raw input data. For the experimental setup, the scaling factor *b* was set at 0.5 to prevent biassing in either direction.

Equation (10) - Moving average filter

$$out = prev - (b\ (prev - raw))$$

Accuracy of real-time displacement measures

Real-time capture of joint displacement values carries risk of inaccuracy created by aspects such as processing latency or noisy sensor data. By contrast, offline processing does not carry these risks as displacement values can be obtained through robust, replicable methods. As joint displacement is the critical data used to detect and describe movement (including compensatory movement), it is essential that all values captured in real-time are accurate and reliable. Therefore, to explore the accuracy of displacement values captured in real-time, which in turn dictate the accuracy of movement detection, it is necessary to compare real-time to offline calculated data for error.

To generate a dataset, 10 able bodied study participants were recruited from the Glasgow Caledonian University staff and student community to perform reaching motions facing the Kinect camera. Ethical approval was granted by Glasgow Caledonian University School of Health and Life Sciences ethics committee (reference: HLS/PSWAHS/19/001). Inclusion criteria stated all participants must be 18 years or older, capable of providing informed consent and performing the required upper-limb movements as per self-report. Exclusion criteria stated participants must not have a history of or any ongoing neurological pathologies, musculoskeletal pathologies, or any consideration which would potentially influence the performance of upper body movements. Written informed consent was obtained from all participants prior to data capture. Participants (female = 3, male = 7) were aged between 19 and 51 years of age (mean = 31) and of mixed arm dominance (left = 2, right = 8, as per self-report).

The Kinect camera was set to operate at a sampling frequency of 30 Hz. From an upright, seated position, participants were tasked with reaching between three target points. The target heights and positions were adapted based on each participant's anthropometry. Target one was placed at the height of each participant's shoulder (min = 95 cm, max = 108 cm, mean = 103 cm) approximately 30 cm in front of the dominant hand. Targets two and three were placed at shoulder height, shoulder width apart at arm's length on the ipsilateral and contralateral side of the dominant hand. Participants alternated reaching patterns between target one and two, then target one and three. A total of 2813 reaching motions (mean = 281 per participant) were captured in real-time and exported from the game engine for analysis. From these data, two categories were established and compared for error:

1. Movement (TF, TR, SR, FR) calculated through offline analysis
2. Movement (TF, TR, SR, FR) calculated through real-time capture

For the offline data category, raw trajectory data which bypasses the real-time moving average filter was exported from the game engine. These data are filtered using a second order Butterworth filter at 2 Hz and a local minima and maxima search on the wrist trajectory is employed within MATLAB[41] to determine the beginning and end of each reach cycle. A time stamp is established and used to locate the position of all other joints at this time. These data are used to establish joint displacement, allowing for contribution to reach to be calculated for the offline data category. To establish the real-time data category, the real-time generated timestamps which denote the beginning and end of each reach cycle are imported into MATLAB and used to locate the relevant positions of the wrist, elbow, shoulder, and trunk. From this, displacement values for each joint are calculated and contribution to reach is assessed. The data from all reach cycles for each feature in each category are summed, and the mean value for each movement type is obtained. To establish error, a simple calculation was performed (Equation (11)).

Equation (11) - Error calculation

$$error = |real\ time\ mean - offline\ mean|$$

Table 3 shows reported mean values for all reach features using real-time capture and offline analysis. The highest mean error was observed to be 1.71% for trunk forward inclination, with lowest error at 0.51% for trunk rotation. These results indicate that although differences may exist between real-time and offline measurement, the low error indicates the real-time collection of displacement measures is comparable to offline methods.

Discussion

The algorithm presented in this paper analyses reaching motions in real-time and provides quantitative outcomes which aim to describe task performance in the sagittal plane. The algorithm makes steps towards a more comprehensive description of upper limb motion than previous work[19] through the inclusion of shoulder rotation and offers real-time operation through a low-cost, markerless motion capture camera. The real-time collection of displacement

Table 3. Real-time algorithm versus offline analysis error.

Movement	Real-time (mean %)	Offline (mean %)	Absolute error
Trunk forward (TF)	17.94	19.65	1.71
Shoulder rotation (SR)	16.33	15.08	1.25
Forward reach (FR)	47.43	46.24	1.19
Trunk rotation (TR)	19.44	18.93	0.51

measures was compared to offline measurement and shows minimal error, indicating movements can be detected in real-time with sufficient accuracy for clinical practice. Microsoft Kinect was employed as an input device. However, the algorithm itself is independent of the sensor and can operate with any input device which can capture the positions of the wrist, elbow, shoulder, and spine, allowing the algorithms to be integrated into other game systems designed for upper limb rehabilitation. Research into rehabilitation games often focuses on engagement[12] or effects via pre and post intervention testing. A systematic review of randomised controlled trials which utilised games as an intervention aimed to establish if a significant difference in motor function was seen when stroke survivors used game systems that were specifically designed for upper limb rehabilitation versus systems which were not designed with rehabilitation in mind.[15] The review concluded that systems specifically designed for rehabilitation had a greater effect on recovery than those that did not. Only one game system satisfied all rehabilitation criteria outlaid by the authors, indicating there is a significant gap in research in this area.

The four movement types (TF, TR, SR, FR) are detected and quantified in relation to each other. In complex motions such as reaching, this can be beneficial as there may be a multitude of different motions which combine to form the complete action. Therefore, these motions should not be considered in isolation, but as intrinsically linked features which combine to describe task performance. Recent work has developed multi-label classification algorithms with some success.[28] However, the user and health professional are only informed that each compensatory motion was present. This results in ambiguous output as no insight into the amount of compensation or the relationship between each motion is reported. By contrast, the algorithm presented in this paper detects each movement and furthers this by quantifying and aggregating the contribution of all features to provide an overview of the complete motion. This high-level aggregate information is essential to guide the motor learning process and thereby optimise rehabilitation outcomes from playing the game.

Analysis of compensatory movement has been investigated, yet there currently remains no consensus on the most effective means of detecting and quantifying these movements within a game context. Common metrics seen in the literature are joint angles[42-44] which can be prone to error in consumer level depth cameras[38] and require the interpretation of numerous angles to describe a complex motion such as reaching. Therefore, it could be argued that three-dimensional reconstruction of joint angles - although the gold standard in biomechanics - is impractical and arguably yields an excessive amount information for use within a home-based rehabilitation game, with minimal supervision from a therapist. To offer a pragmatic alternative that can be implemented in routine rehabilitation practice, the algorithm presented simplifies all motion and allocates a single percentage value for each movement type. This negates the need to interpret multiple joint angles and allows for simple, graphical presentation of data, examination of longitudinal trends, and easily adapted in-game feedback systems to motivate and engage the stroke survivor. Although this method does not produce the level of detail and accuracy seen in marker-based kinematic analysis, the simplified representation and dimensionless quantification may be sufficient for home use due to the accessible means of data presentation, low costs, and reasonable tracking accuracy found in consumer level motion capture cameras.[39] However, further research is required to assess the feasibility of such systems.

The algorithm tailors to the user's current motor function by establishing their comfortable level of elbow extension. This calibration can be performed in seconds, allowing the system to evolve with the ability of the user and requires no downtime for recalibration, permitting continuous use which is appropriate and safe. Moreover, as the algorithm utilises the relative positioning of joints for all calculations, the detection of new motions can be easily incorporated as each calculation need only consider which joints are relevant to the specific measure and how the relationship between these joints quantifies the motion. As these calculations are all relative to the comfortable level of elbow extension, they evolve with the user, meaning all output measures and in-game feedback related to movement is accurate and evolves as motor function improves.

Game-based rehabilitation systems rarely offer comprehensive motion analysis and clinically relevant metrics related to compensatory tasks.[22] Consequently, there is currently no standard toolkit for the detection of compensatory movement which offers an easy to understand, gameplay independent indicator of task performance. To advance development in this area, it is intended that the presented algorithms might be used as a basis for future development of game-based repetitive practice systems involving reaching tasks. Furthering the work presented in this paper, the system will be expanded to include a larger variety of compensatory motions (such as elevation of the shoulder girdle, trunk lateral flexion, and assisting with the less affected arm) then seek to package all algorithms within a distributable toolkit for easy integration into home-based rehabilitation systems. Future work should explore robust testing of the algorithms with stroke survivors with upper limb impairment to assess the accuracy and suitability for home-based, unsupervised use.

Conclusion

This paper presented an algorithm which analyses cyclical upper limb reaching motions in real-time and segments the action into its component parts with reference to trunk forward inclination, trunk rotation, shoulder rotation and forward reach. The real-time operation of the algorithm was tested for accuracy by comparing data collected in real-time to offline analysis of trajectory data. Results indicate that the algorithm can operate in real-time with accuracy, showing a maximum error of 1.71%. Future work will extend the system to provide further analysis of trunk and upper limb motion and create a distributable toolkit for the detection and quantification of compensatory movements.

Acknowledgements

All study participants, Axel Horbach, Ben Stansfield, Danny Rafferty at Glasgow Caledonian University

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was internally funded by a Glasgow Caledonian University studentship

Guarantor

EA

Contributorship

EA developed algorithms, performed data analysis, and contributed to the development of the manuscript. DK and FVW were responsible for project supervision and contributed to the development of the manuscript. AH recruited participants, contributed to data collection, and gained ethical approval.

ORCID iD

Edward Averell https://orcid.org/0000-0001-8804-5868

References

1. Van Wijck F, Knox D, Dodds C, et al. Making music after stroke: using musical activities to enhance arm function. *Ann N Y Acad Sci* 2012; 1252(1): 305–311.
2. Averell E and Knox D. A rhythm-based game for stroke rehabilitation. In: *Audio Engineering Society Conference. : 2019 AES International Conference on Immersive and Interactive Audio*. York: AES, paper no. 21.
3. Faria-Fortini I, Michaelsen SM, Cassiano JG, et al. Upper extremity function in stroke subjects: relationships between the international classification of functioning, disability, and health domains. *J Hand Ther* 2011; 24(3): 257–265.
4. Haghgoo HA, Pazuki ES, Hosseini AS, et al. Depression, activities of daily living and quality of life in patients with stroke. *J Neurol Sci* 2013; 328: 87–91.
5. Pollock A, Farmer SE, Brady MC, et al. Interventions for improving upper limb function after stroke. *Cochrane Database Syst Rev* 2014; 11, CD010820,
6. French B, Thomas LH, Coupe J, et al. Repetitive task training for improving functional ability after stroke. *Cochrane Database Syst Rev* 2016; 11, CD006073,
7. Cramer SC, Sur M, Dobkin BH, et al. Harnessing neuroplasticity for clinical applications, *Brain* 2011; 134(6): 1591–1609.
8. Scottish Government. *Stroke Improvement Plan*, https://www.strokeaudit.scot.nhs.uk/Publications/Main.html (2020, accessed 10 March 2020).
9. Poltawski L, Allison R, Briscoe S, et al. Assessing the impact of upper limb disability following stroke: a qualitative enquiry using internet-based personal accounts of stroke survivors. *Disabil Rehabil* 2016; 38(10): 945–951.
10. Stroke Association. *Stroke recoveries at risk*, https://www.stroke.org.uk/stroke-recoveries-at-risk-report (2020, accessed 25 January 2021).
11. Thomson K, Pollock A, Bugge C, et al. Commercial gaming devices for stroke upper limb rehabilitation: The stroke survivor experience. *J Rehabil Assist Technol Eng* 2020; 7: 1–13.
12. Flores E, Tobon G, Cavallaro E, et al. Improving patient motivation in game development for motor deficit rehabilitation. In: *Proceedings of the 2008 international conference on advances in computer entertainment technology*, Yokohama, Japan, 2008, pp. 381–384.
13. Laver KE, Lange B, George S, et al. Virtual reality for stroke rehabilitation*Cochrane Database Syst Rev* 2017; 11: CD008349.
14. Barrett N, Swain I, Gatzidis C et al. The use and effect of video game design theory in the creation of game-based systems for upper limb stroke rehabilitation. *J Rehabil Assist Technol Eng* 2016; 3: 205566831664364. DOI: 10.1177/2055668316643644.

15. Maier M, Rubio Ballester B, Duff A, et al. Effect of specific over nonspecific VR-based rehabilitation on poststroke motor recovery: a systematic meta-analysis. *Neurorehabil Neural Repair* 2019; 33(2): 112–129.
16. Cirstea MC and Levin MF. Compensatory strategies for reaching in stroke. *Brain* 2000; 123(5): 940–953.
17. Roby-Brami A, Feydy A, Combeaud M, et al. Motor compensation and recovery for reaching in stroke patients. *Acta Neurol Scand* 2003; 107(5): 369–381.
18. Murphy MA, Willén C and Sunnerhagen KS. Kinematic variables quantifying upper-extremity performance after stroke during reaching and drinking from a glass. *Neurorehabil Neural Repair* 2011; 25(1): 71–80.
19. Aprile I, Rabuffetti M, Padua L, et al. Kinematic analysis of the upper limb motor strategies in stroke patients as a tool towards advanced neurorehabilitation strategies: a preliminary study. *Biomed Res Int* 2014: 1–8, DOI: 10.1155/2014/636123.
20. Levin MF, Kleim JA and Wolf SL. What do motor "recovery" and "compensation" mean in patients following stroke? *Neurorehabil Neural Repair* 2009; 23(4): 313–319. DOI: 10.1177/1545968308328727.
21. Raghavan P. Upper limb motor impairment after stroke. *Phys Med Rehabil Clin N Am* 2015; 26(4): 599–610.
22. Tamayo-Serrano P, Garbaya S and Blazevic P. Gamified in-home rehabilitation for stroke survivors: analytical review. *Int J Serious Games* 2018; 5(1). DOI: 10.17083/ijsg.v5i1.224.
23. Tormene P, Giorgino T, Quaglini S, et al. Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation. *Artif Intelligence Medicine* 2009; 45(1): 11–34.
24. Su CJ. Personal rehabilitation exercise assistant with kinect and dynamic time warping. *Int J Inf Educ Technol* 2013; 3(4): 448–454.
25. Cary F, Postolache O and Girão PS. Kinect based system and artificial neural networks classifiers for physiotherapy assessment. In: 2014 IEEE International Symposium on Medical Measurements and Applications. Lisbon: IEEE, 2014, pp. 1–6.
26. Vakanski A, Jun HP, Paul D, et al. A data set of human body movements for physical rehabilitation exercises. *Data* 2018; 3(1): 2. DOI: 10.3390/data3010002.
27. Cai S, Li G, Zhang X, et al. Detecting compensatory movements of stroke survivors using pressure distribution data and machine learning algorithms. *J Neuroeng Rehabil* 2019; 16(1): 131. DOI: 10.1186/s12984-019-0609-6.
28. Kashi S, Feingold-Polak R, Lerner B, et al. A machine-learning model for automatic detection of movement compensations in stroke patients. *IEEE Trans Emerging Top Comput* 2020; 9: 1234–1247. DOI: 10.1109/TETC.2020.2988945.
29. Woodman S, Hiden H, Turner M, et al. *Monitoring of upper limb rehabilitation and recovery after stroke: an architecture for a cloud-based therapy platform*. In: 2015 IEEE 11th International Conference on e-Science, August 2015, Germany, pp. 381–390.
30. Joo LY, Yin TS, Xu D, et al. A feasibility study using interactive commercial off-the-shelf computer gaming in upper limb rehabilitation in patients after stroke. *J Rehabil Med* 2010; 42(5): 437–441.
31. Microsoft Corporation. *Kinect for Windows*, https://developer.microsoft.com/en-us/windows/kinect (2020, accessed 05 August 2020).
32. Shin DJ, Kim MS, Song W, et al. Implementation of sitting posture monitoring system with kinect. In: Advanced Multimedia and Ubiquitous Engineering, May 2017, Singapore, pp. 144–150.
33. Brokaw EB, Eckel E and Brewer BR. Usability evaluation of a kinematics focused Kinect therapy program for individuals with stroke. *Technol Health Care* 2015; 23(2): 143–151.
34. Foreman MH and Engsberg JR. A virtual reality tool for measuring and shaping trunk compensation for persons with stroke: Design and initial feasibility testing. *J Rehabil Assist Technol Eng* 2019; 6: 205566831882367. DOI: 10.1177/2055668318823673.
35. Dukes PS, Hayes A, Hodges LF, et al. Punching ducks for post-stroke neurorehabilitation: System design and initial exploratory feasibility study. In: 2013 IEEE Symposium on 3D User Interfaces. Orlando, 2013, pp. 47–54.
36. Da Gama A, Chaves T, Figueiredo L, et al. Guidance and movement correction based on therapeutics movements for motor rehabilitation support systems. In: 2012 14th symposium on virtual and augmented reality, May 2012, Washington, pp. 191–200.
37. Da Gama AE, de Menezes Chaves T, Fallavollita P, et al. Rehabilitation motion recognition based on the international biomechanical standards. *Expert Syst Appl* 2019; 116: 396–409.
38. Napoli A, Glass S, Ward C, et al. Performance analysis of a generalized motion capture system using microsoft kinect 2.0. *Biomed Signal Process Control* 2017; 38: 265–280.
39. Otte K, Kayser B, Mansow-Model S, et al. Accuracy and reliability of the kinect version 2 for clinical measurement of motor function. *PLoS One* 2016; 11(11): e0166532. DOI: 10.1371/journal.pone.0166532.
40. Epic Games. Unreal Engine, https://www.unrealengine.com (2020, accessed 03 September 2020).
41. Mathworks. Matlab, https://uk.mathworks.com/products/matlab.html (2020, accessed 03 September 2020).
42. Kuster RP, Heinlein B, Bauer CM, et al. Accuracy of KinectOne to quantify kinematics of the upper body. *Gait & Posture* 2016; 47: 80–85.
43. Reither LR, Foreman MH, Migotsky N, et al. Upper extremity movement reliability and validity of the Kinect version 2. *Disabil Rehabil Assistive Technol* 2018; 13(1): 54–59.
44. Scano A, Molteni F and Molinari Tosatti L. Low-cost tracking systems allow fine biomechanical evaluation of upper-limb daily-life gestures in healthy people and post-stroke patients. *Sensors* 2019; 19(5): 1224. DOI: 10.3390/s19051224.