Received August 26, 2017, accepted October 1, 2017, date of publication October 6, 2017, date of current version October 25, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2759801

INVITED PAPER

Rule-Based Human Motion Tracking for Rehabilitation Exercises: Realtime Assessment, Feedback, and Guidance

WENBING ZHAO[1], (Senior Member, IEEE), M. ANN REINTHAL[2], DEBORAH D ESPY[2], AND XIONG LUO[3], (Member, IEEE)

[1]Department of Electrical Engineering and Computer Science, Cleveland State University, Cleveland, OH 44115, USA
[2]School of Health Sciences, Cleveland State University, Cleveland, OH 44115, USA
[3]School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China

Corresponding author: Wenbing Zhao (w.zhao1@csuohio.edu)

This work was supported in part by the Undergraduate Summer Research Award from Cleveland State University, in part by the Graduate Faculty Support Award, and in part by the Faculty Research Development Award.

ABSTRACT In this paper, we report the design and implementation of a Kinect-based system for providing automated realtime assessment, feedback and guidance to users who are practicing rehabilitation exercises at home without the supervision of physical therapists. The foundation for the system is a rule-based framework that can be used to assess in realtime the quality and quantity of the exercises performed by the user. We demonstrate the capability of the rule-based framework by showing the detailed rules for three common rehabilitation exercises, including bowling, hip abduction, and sit to stand. To test its usability and accuracy, we have used the system in a human subject study with eight healthy users. The results show that with proper empirical parameters in the rules, the performance of these exercises can be reliably assessed in realtime.

INDEX TERMS Rule-based human motion assessment, microsoft Kinect, human activity recognition, rehabilitation exercises, finite state machine.

I. INTRODUCTION

Physical exercises constitute an essential part of preventive and rehabilitative medicine [1]. Typically, a therapist would prescribe a rehabilitation program for a patient to help treat symptoms and speed up the recovery process. The patient is expected to perform the prescribed exercises at home daily with many repetitions. The current state of the practice is far from adequate. In many cases, printed materials with simple illustrations and purposes of the prescribed exercises were given to a patient. In recent years, videos (or links to videos posted on the Web) of the prescribed exercises are provided to the patient in addition. The paper-only materials are apparently not desirable. Even with the help of the videos, patients may still be left wondering if he/she is doing the exercises correctly. Another important impeding factor is that such materials are not engaging enough to get patients to perform the prescribed exercises daily at home without the supervision of any therapist, let alone to perform them correctly with sufficient repetitions.

It is well known that gamification can help engage people in physical exercises [2]. However, current games or game-like programs designed for rehabilitation are typically limited to repetition count and scoring. Some cutting edge research prototypes are capable of providing more specific feedback, such as [3] where only the violations in static axes are detected. While this is a good progress forward, it is still not desirable because physical therapeutic exercises typically involve dynamic movements. Making sure that a patient is performing the prescribed exercises exactly as specified is crucial for a speedy recovery, and performing them incorrectly may in fact cause injuries to the patient [4], [5]. Hence, a good assistive technology for helping patients to carry out the prescribed exercises correctly and adequately at home without supervision must accurately detect all violations of the physical movements performed by patients in *realtime* and provide specific feedback so that they know what they did wrong and make corrections.

Assistive technology for therapeutic exercises should also provide a mechanism for customization. Depending on the severity of the health problems of the patient, an exercise would need to be customized to fit the patient's condition. This applies not only to different patients, but also the same patient. During the course of the recovery, the rules for a prescribed exercise might need to be changed as well. For example, for a stroke patient, he/she could hardly move his/her arm initially, and as he/she progresses in recovery, he/she would be able to move his/her arms more freely. If the bowling exercise is prescribed, the range of motion should be quite small in the beginning and it should be increased gradually over time. Furthermore, the tolerance on movement trajectory should also be gradually tightened up.

In this paper, we present a rule-based framework for specifying therapeutic exercises. Because the exercises are described in terms of rules, the specification also serves as the basis for realtime feedback where the patient will be informed of exactly which rule is violated during a repetition. The parameters used in the rules, such as the beginning and ending pose orientations, as well as the tolerance values, can be easily adjusted dynamically. Subsequently, we describe the design and implementation of an avatar-based guidance system that incorporates the rule-based framework. The usability of the system is validated via a human subject study with eight healthy subjects.

II. RELATED WORK

There are primarily two approaches to human activity recognition: (1) learning based and (2) rule based. In the learning based approach, labeled training data are first obtained, and then they are used either directly or indirectly via a predefined model to classify a newly acquired activity [6]–[13]. In the rule based approach, the activity is first described in terms of a set of rules and an unknown activity is classified based on the rules defined for each possible activity.

Unlike the rule-based approach, the learning-based approach can be used to classify an arbitrary activity without the need to fully describe it, provided that there are sufficient training data for the activity. The learning-based approach obviously has advantages over the rule-based approach in terms the cost of describing an activity because in the rule-based approach, each activity must be painstakingly fully described by rules with appropriate parameters, which requires intimate knowledge about every activity involved. On the other hand, this strength in generality also constitutes the weakness of the learning-based approach in the context of guiding a patient on performing therapeutic exercises correctly because it is difficult to provide specific feedback to the patient regarding exactly what was performed wrong. Furthermore, it is also difficult to obtain sufficient training data in cases when an exercise must be customized with gradually changing range of motion and tolerance values.

Because we take the rule-based approach, we only outline related work that follows the same approach. General-purpose rules have been proposed for classifying hand gestures [14] and whole body activities [15]. In the latter [15], a Gesture Description Language (GDL) is introduced to describe common activities. Each activity (or gesture) is identified by a set of rules. There are two different types of rules, the basic rules and the final rule. Each basic rule is defined in terms of of one or more key frames each consisting of a set of joint positions. The final rule is defined in terms of a sequence of basic rules. The use of key frames as the building blocks in [15] has its limitation in that it cannot define rules that rely on the entire trajectory of a gesture. Furthermore, no guideline on how to identify key frames was provided in [15].

It is worth noting that in the context of therapeutic exercise monitoring and guidance, rules defined for an exercise are primarily used to determine the quality of movements rather than to recognize which exercise the patient is performing because the exercise to perform is considered to be known from the context. Therefore, it is unnecessary for the rules to completely describe an exercise. As a result, the number of rules is small and they are usually defined in terms of joint angles. For example, in [16] and [17], gait retraining rules are defined in terms of the trunk flexion angle, trunk lean angle, and the distance a set of joints for postural control traverse. In [18], the rules for sit-to-stand and squat exercises are defined in terms of the knee angle and the ankle angle, and the rules for shoulder abduction/adduction are defined in terms of the shoulder angle. In [19], the rules for knee rehabilitation exercises are defined in terms of the knee angle. In [20], the rules for the sit-to-stand exercise are expressed in terms of the minimum hip angle and the movement smoothness of the head.

The construction and expression of rules in our framework are influenced by [14], where a hand gesture is defined by a sequence of movement segments in terms of a monotonically increasing or decreasing key parameter, such as as the angle between two fingers. Each segment is referred to as a monotonic segment. In our framework, we also also monotonic segments to define dynamic rules. The frames that delineate two consecutive segments are refereed to as reference configurations. Unlike [14], where only dynamic rules were defined, we also define invariance rules and static rules [21], [22]. The invariance rules specify the rules by which the movement must abide by throughout the entire repetition. The static rules specify the poses of segments that must remain stationary. The invariance rules and static rules are critical for rehabilitation exercises, but they may not be important for gesture/activity recognition.

III. RULE-BASED ASSESSMENT FRAMEWORK FOR REHABILITATION EXERCISES

Rehabilitation exercises are assessed via three sets of rules, rules for dynamic rules (dynamic rules for short), rules for static poses (static rules for short), and rules for movement invariance (invariance rules for short):

- Dynamic rules. A dynamic rule defines a sequence of consecutive key positions (*i.e.,* reference configurations) of a moving body segment for each repetition.
- Static rules. A static rule specifies the position and/or orientation of a key body segment that should remain stationary at during each repetition.

- Invariance rules. Each invariance rule describes a condition that a moving body segment must meet during each repetition.

All rules are described in terms of one or more reference configurations. It can be defined in a number of ways depending on the context. It could be in terms of the joint angle in between two adjacent body segments, such as the angle between a moving leg and the the stationary one. It may be in terms of the orientation with respect to one of the anatomical planes, such as the frontal, sagittal, or transverse plane. It may also be defined in the form of the distance between some joints or relative positions of different joints (particular for static rules).

A. ENCODING OF RULES

Rules are encoded using the eXtensible Markup Language (XML) for their readability and extensibility. Listing 1 shows a template for encoding the rules. The rules start with an ExerciseRules element where the exercise name should be given as an attribute for identification purpose. This is followed by a list of dynamic rules enclosed in a DynamicRules element, one or more static rules in a single StaticRules element, and one or more invariance rules in an InvarianceRules element.

```
<ExerciseRules  Name="Exercise Name">
    <DynamicRules> ... </DynamicRules>
    <StaticRules> ... </StaticRules>
    <InvarianceRule> ... </InvarianceRule>
</ExerciseRules>
```

Listing 1. **The template structure on rule encoding for an exercise.**

The DynamicRules element consists of one or more DynamicRule elements. Each DynamicRule element consists of two or more Configuration elements, as shown in Listing 2. If there is only a single DynamicRule, then the structure can be compacted to have the DynamicRule directly under the ExerciseRules element.

```
<DynamicRule>
    <Configuration> ... </Configuration>
    <Configuration> ... </Configuration>
    ...
    <Configuration> ... </Configuration>
</DynamicRule>
```

Listing 2. **Encoding of dynamic rule.**

The StaticRules element and the Invariance element are rather similar to the DynamicRule element. Both consist of a list of Configuration elements, as shown in Listing 3.

In our framework, three types of Configuration elements are used, which are shown in Listings 4, 5, and 6. Each Configuration element encodes a reference configuration and it all starts with a Type element so that the element can be parsed properly. The JointAngle type defines a configuration using the joint angle between two adjacent body segments. The CenterJoint element defines the common joint of the

```
<StaticRules/InvarianceRules>
    <Configuration> ... </Configuration>
    <Configuration> ... </Configuration>
    ...
    <Configuration> ... </Configuration>
</StaticRules/InvarianceRules>
```

Listing 3. **Static/Invariance rule encoding.**

```
<Configuration>
    <Type>"JointAngle"</Type>
    <CenterJoint>"JointName"</CenterJoint>
    <DownstreamJoint>"JointName"</DownstreamJoint>
    <UpstreamJoint>"JointName"</UpstreamJoint>
    <Angle>"ExpectedAngleValue"</Angle>
    <MaxAngleDeviation> "..." </MaxAngleDeviation>
    <Duration>"ExpectedDurationValue"</Duration>
    <MaxDurationDeviation> "..." </MaxDurationDeviation>
</Configuration>
```

Listing 4. **A configuration for joint angle.**

two segments. The DownstreamJoint and UpstreamJoint elements specify the other endpoints of the two segments. The Angle and MaxAngleDeviation element are self-explanatory. The tolerance value in the MaxAngleDeviation element is determined heuristically and empirically. When determining this parameter, the following factors should be considered:

- The tolerance value must not lead to the overlapping of the current configuration with any other configuration to avoid confusion.
- The tolerance value must not result in an unsafe posture for the user.

The JointDistance type defines a configuration using distance (as indicated in the Distance element) between two joints, shown in Listing 5. One joint is a moving joint (*i.e.*, the MovingJoint element), and a stationary joint (*i.e.*, the StationaryJoint element). The MaxDistDeviation element defines the tolerance value.

```
<Configuration>
    <Type>"JointDistance"</Type>
    <MovingJoint>"JointName"</MovingJoint>
    <StationaryJoint>"JointName"</StationaryJoint>
    <Distance>"ExpectedValue"</Distance>
    <MaxDistDeviation> "..."</MaxDistDeviation>
    <Duration>"ExpectedDurationValue"</Duration>
    <MaxDurationDeviation> "..." </MaxDurationDeviation>
</Configuration>
```

Listing 5. **A configuration for distance between two joints.**

The BoneOrientation type specifies a configuration using the orientation of a body segment. The body segment is defined using two joints, DownstreamJoint and UpstreamJoint. To describe the segment orientation, we decide to adapt from the spherical coordinates for the 3-dimensional space. To define a set of spherical coordinates, we need to choose an origin, which we decide to use the DownstreamJoint, a plane that contains the origin, and two mutually perpendicular axes passing through the origin. Because many rehabilitation exercises involve the

```
<Configuration>
    <Type>"BoneOrientation"</Type>
    <DownstreamJoint>"JointName"</DownstreamJoint>
    <UpstreamJoint>"JointName"</UpstreamJoint>
    <Plane>"PlaneName"</Plane>
    <Axis>"Axis Name"</Axis>
    <AlphaAngle>"ExpectedAngleValue"</AlphaAngle>
    <BetaAngle>"ExpectedAngleValue"</BetaAngle>
    <MaxAngleDeviation> "..." </MaxAngleDeviation>
    <Duration>"ExpectedDurationValue"</Duration>
    <MaxDurationDeviation> "..." </MaxDurationDeviation>
</Configuration>
```

Listing 6. A configuration for bone orientation.

movement of certain body segment along some anatomical plane, we encourage the user to use the most relevant anatomical plane as the plane required by the set of spherical coordinates. One of two axes should be the axis that is perpendicular to the anatomical plane. The user has freedom to choose the other axis that lies within the anatomical plane.

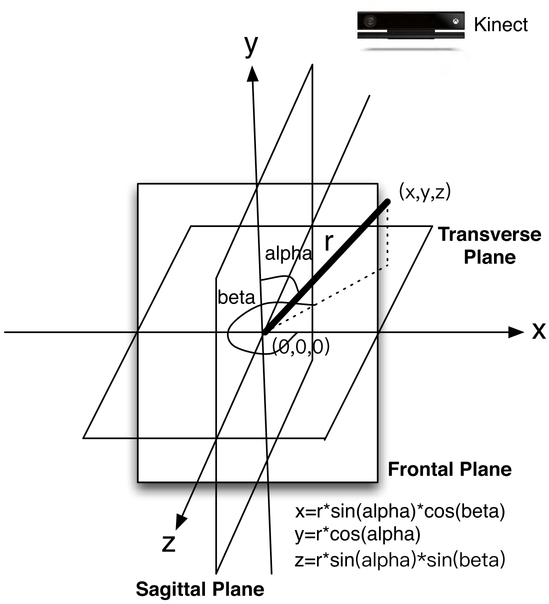


FIGURE 1. Spherical coordinates in defining the bone orientation.

The definition of the planes and the axes used in defining the orientation of a body segment is provided in Figure 1. We choose to define a coordinates system that is similar to that used by Microsoft Kinect software development kit. If the Kinect sensor is positioned up right without any tilt, and the user is standing facing the Kinect sensor, then the frontal plane would coincide with the plane determined by the x-y axes, and the z-axis would be pointing away from the Kinect sensor to the back of the user. The transverse plane would coincide with the floor (assuming that the floor is strictly flat) and are determined by the x-z axes. the sagittal plane is determined by the y-z axes.

Given a body segment, where the DownstreamJoint has a Cartesian coordinates of (0,0,0), and the UpstreamJoint has a Cartesian coordinates of (x,y,z), the spherical coordinates of the UpstreamJoint are $(r, alpha, beta)$ according to the setup used in Figure 1 where the plane of interest is the transverse plane, and the two axes used are x and y axes. Here r is the length of the body segment. The angle *alpha* is defined as the angle between the body segment and the y-axis. The angle *beta* is defined to be the angle between the x and y axes. Consequently, the relationship between the Cartesian coordinates and the spherical coordinates for the UpstreamJoint is: $x = r * sin(alpha) * cos(beta)$, $y = r * cos(alpha)$, $z = r * sin(alpha) * sin(beta)$.

To define the orientation of a body segment, the length of the segment r is not relevant and only the plane, the two axes, and the two angles are important. Because one of the axes must be the axis that is perpendicular to the plane, it can be inferred from the plane itself. Hence, that axis is not necessary either. The reference configuration would need to define the plane, one axis, and the two angles, as shown in Listing 6. The valid value for the Plane element includes "Frontal", "Sagittal", and "Transverse". The valid values for the Axis element includes "X", "Y", and "Z". When used for invariance rule to limit the movement of a body segment within an anatomical plane, the alpha value will be 0 with no beta value and no second axis specified, and the plane should be that anatomical plane.

When used to describe a dynamic rule, all three types of Configuration elements may contain the Duration and MaxDurationDeviation elements, which specify the desirable duration range of the current monotonic segment starting with the current configuration.

IV. REALTIME MOTION TRACKING

The quality of an exercise is assessed by examining the skeleton frames provided by the Kinect sensor based on the rules for the current rehabilitation exercise. Typically, there are two levels of analysis based on the defined rules: the frame level and the system level. All rehabilitation exercises are repetitive. Hence, the unit of quality assessment is one repetition and the user is informed about the quality of the current repetition at the end of the repetition. However, feedback is provided to the user while he/she is doing the exercise in realtime even before the current repetition is completed. For example, on detection of the violation of one or more rules, some form of feedback (such as visual cue) is immediately provided to the user. The repetition is determined by applying the dynamic rules.

To deal with occasional jitters and measurement errors, raw data can be filtered using some standard filter. In our experiments, we find that the complementary filter works fairly well because it offers a good compromise in combining slow and fast moving signals. Given a sequence of raw data of certain variable, such as a joint angle, raw_n, where $n = 1, 2, 3, \ldots$, and the corresponding filtered data, $filtered_n$, is determined by the following formula:

$$filtered_n = \alpha * filtered_{n-1} + (1 - alpha) * raw_n. \quad (1)$$

Here α is a smoothing parameter between 0 to 1.

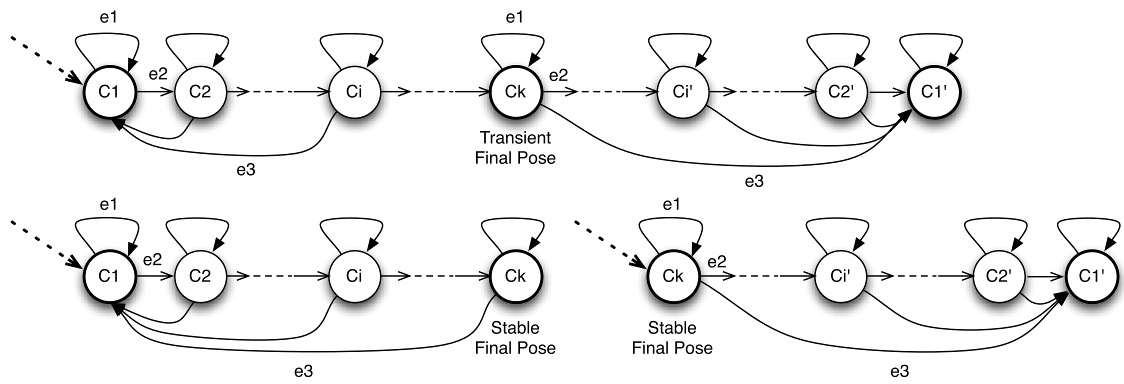


FIGURE 2. Two types of finite state machines for the dynamic rule, one with a transient final pose (top), and the other with a stable final pose (bottom).

For static rules and invariance rules, the detection of a single (filtered) frame violation would render the current repetition unacceptable. The tracking of repetition and the quality of performance with respect to dynamic rules are much more complicated and we will elaborate in detail below.

Realtime motion tracking is guided by a finite state machine, as shown in Figure 2. The number of states are identical to that of the monotonic segments, which in turn is the number of reference configurations. The first challenge in motion tracking is to determine when a repetition is started. To do so, we depends on the first configuration in the dynamic rule. This configuration corresponds to the initial state in the state machine as indicated by $C1$.

Because of the repetitive nature of rehabilitation exercises, one repetition contains two back-to-back mirrored activities. For example, for the hip-abduction exercise, one repetition starts with a hip-abduction activity and is followed by a hip-adduction activity. One repetition of the sit-to-stand exercise starts with a sit-to-stand activity and is followed by the mirroring stand-to-sit activity. These two mirroring activities of each repetition lead to another interesting phenomenon: the final pose of the first activity may be quite different. The final pose can transient, or it can be a stable pose. For example, at the end of the hip-abduction activity, the abducting leg reaches out to the maximum hip angle, which is a transient pose because it is difficult for a user to remain at this pose.

In contrast, the first activity of the sit-to-stand exercise ends with the user standing straight up, which is a stable final pose because the person could stay comfortably in that pose for much longer time than that of a transient pose before proceeding to the stand-to-sit activity. For exercises with a transient pose, it makes sense to track the exercise as a single state machine. For exercises with a stable pose, it is best to model the exercise as two separate finite state machines. The two types of finite state machines are illustrated in Figure 2.

We start by explaining the finite state machine specification for exercises that has a transient final pose. An exercise is modeled to have k reference configurations as defined in the dynamic rules. Consequently, there are $2k-1$ states in the finite state machine for the exercise. Each state corresponds to a reference configuration defined. Hence, we reuse the same symbol C_i to refer to the reference configuration as well as the corresponding state. The $2k-1$ states are represented as $C_1, C_2, \ldots, C_{k-1}, C_k, C^{;}_{k-1}, \ldots, C^{;}_2, C^{;}_1$, where $C^{;}_i$ is the mirror motion segment of C_i. The machine enters a state C_i on detecting the corresponding reference configuration and will stay in that state until the detection of the next reference configuration as shown in Figure 2.

For exercises with a stable final pose, two separate finite state machines are used to track each repetition. The state transition is identical to that in the finite state machine with a transient final pose.

In practice, a patient may not execute an exercise exactly as described. As such, it is inadequate to compare against the next reference configuration according to the finite state machine because the expected reference configuration may not be satisfied. Therefore, we must track each actual monotonic segment dynamically as the user is performing the exercise. This is accomplished by tracking the variables defined in the rule (such as the joint angle, the body segment orientation angle, or the distance between two joints). The current monotonic segment ends when the variable reaches a peak (if the variable is increasing) or a valley (if the variable is decreasing).

A user may decide to stop in the middle of a repetition, perhaps because of the feedback received or some other reasons. To allow this to happen, we add a transition from any of the states to the initial state in the finite state machine. Consequently, in addition to the next expected reference configuration, we also compare the current frame against the initial reference configuration.

In summary, as shown in Figure 2, at a state C_i, three events could happen:

- Event $e1$: The current frame does not match the next expected reference configuration. Hence, the finite state machine stays in the current state C_i.
- Event $e2$: The current frame matches the next expected reference configuration. The finite state machine then switches to the next state C_{i+1}.

- Event $e3$: The current frame satisfies the initial configuration. The finite state machine then switches to the initial state.

The mechanism described above has one weakness in that it does not tolerate jitters if they are severe enough to cause the velocity of the variable to change sign, *i.e.,* when the variable is increasing in value and the jitter makes the next frame to appear to be decreasing in value, and vice versa. We propose the following mechanism to cope with the jitter.

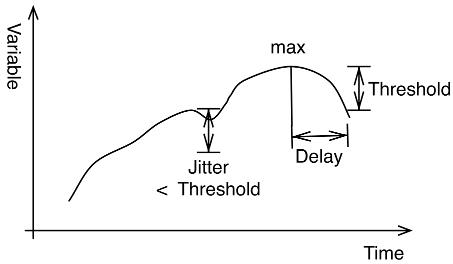


FIGURE 3. **The mechanism to dynamic track when a monotonic segment terminates. As can be seen, if the jitter is smaller than the threshold, our mechanism can survive the jitter.**

The mechanism keeps track of both the maximum and the minimum values of the variable that have been observed in each state. As shown in Figure 3, for a monotonic segment with an increasing (or decreasing) value, it is not terminated until the current value is smaller (or greater) than the last seen maximum/minimum value by a threshold to be resilient to small jitters. The downside of this mechanism is that it would a cause some delay in state transitions. Ultimately, it also causes a slight delay in updating the repetition count on the user interface.

V. CASE STUDIES

In this section, we show how to use our rule-based framework to define rules for three rehabilitation exercises, including bowling, hip abduction, and sit to stand.

A. BOWLING

For the bowling exercise, there is one dynamic rule defining two reference configurations for the movements. As shown in Figure 4, one reference configuration defines the initial position for the bowling exercise where the bowling arm is straightly down or slightly backward from the position, and the other reference configuration is the bowling arm pointing straight forward. This rule not only defines the range of motion of the exercise, but it helps the system to automatically perform repetition count.

There are also two invariance rules. One rule dictates that the bowling arm must be straight, *i.e.,* the angle formed between the lower and upper arms must remain to be 180 degrees the entire time during the exercise. The other rule specifies that the bowling arm must move within the sagittal plane. Bowling does not have any static rule. Listing 7 shows all the rules for this exercise. Both reference configurations in the dynamic rule are defined in terms of the body segment orientation.

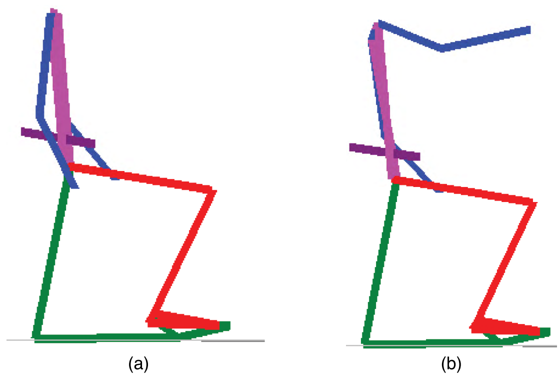


FIGURE 4. **The two reference configurations for the bowling exercise from the sagittal view. (a) The initial pose; (b) the final pose.**

```
<ExerciseRules Name="Bowling">
  <DynamicRule>
    <Configuration>
        <Type>"BoneOrientation"</Type>
        <DownstreamJoint>"JointName"</DownstreamJoint>
        <UpstreamJoint>"JointName"</UpstreamJoint>
        <Plane>"Sagittal"</Plane>
        <Axis>"z"</Axis>
        <AlphaAngle>"0"</AlphaAngle>
        <BetaAngle>"270"</BetaAngle>
        <MaxAngleDeviation> "15" </MaxAngleDeviation>
    </Configuration>
        <Type>"BoneOrientation"</Type>
        <DownstreamJoint>"JointName"</DownstreamJoint>
        <UpstreamJoint>"JointName"</UpstreamJoint>
        <Plane>"Sagittal"</Plane>
        <Axis>"z"</Axis>
        <AlphaAngle>"0"</AlphaAngle>
        <BetaAngle>"180"</BetaAngle>
        <MaxAngleDeviation> "20" </MaxAngleDeviation>
    </Configuration>
  </DynamicRule>
  <InvarianceRule>
    <Configuration>
        <Type>"BoneOrientation"</Type>
        <DownstreamJoint>"RightShoulder"
             </DownstreamJoint>
        <UpstreamJoint>"RightWrist"</UpstreamJoint>
     <Plane>"Sagittal"</Plane>
     <AlphaAngle>"0"</AlphaAngle>
     <MaxAngleDeviation> "20" </MaxAngleDeviation>
    </Configuration>
    <Configuration>
        <Type>"JointAngle"</Type>
        <CenterJoint>"RightElbow"</CenterJoint>
        <DownstreamJoint>"RightWrist"</DownstreamJoint>
        <UpstreamJoint>"RightShoulder"</UpstreamJoint>
        <Angle>"180"</Angle>
        <MaxAngleDeviation>"30" </MaxAngleDeviation>
    </Configuration>
   </InvarianceRule>
</ExerciseRules>
```

Listing 7. **The rules for the bowling exercise.**

B. HIP ABDUCTION

The hip abduction exercise requires that the abducting leg moves away from the body to about 45 degrees in the frontal plane. Usually, the hip-abduction activity is followed by the hip-adduction activity so that the leg would move back to

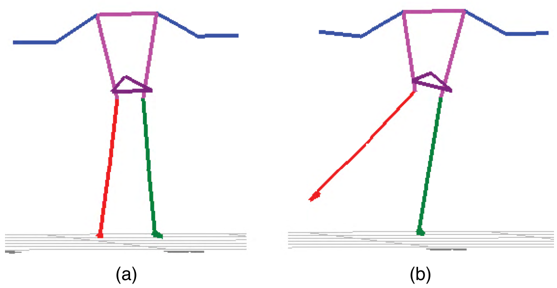


FIGURE 5. **The two reference configurations for the hip abduction exercise from the frontal view. (a) The initial pose; (b) the final pose.**

```
<ExerciseRules Name="Hip Abduction">
  <DynamicRule>
    <Configuration>
        <Type>"JointAngle"</Type>
        <CenterJoint>"HipCenter"</CenterJoint>
        <DownstreamJoint>"RightAnkle"</DownstreamJoint>
        <UpstreamJoint>"LeftAnkle"</UpstreamJoint>
        <Angle>"0"</Angle>
        <MaxAngleDeviation> "15" </MaxAngleDeviation>
    </Configuration>
    <Configuration>
        <Type>"JointAngle"</Type>
        <CenterJoint>"HipCenter"</CenterJoint>
        <DownstreamJoint>"RightAnkle"</DownstreamJoint>
        <UpstreamJoint>"LeftAnkle"</UpstreamJoint>
        <Angle>"45"</Angle>
        <MaxAngleDeviation> "10" </MaxAngleDeviation>
    </Configuration>
  </DynamicRule>
  <InvarianceRule>
    <Configuration>
        <Type>"BoneOrientation"</Type>
        <DownstreamJoint>"HipCenter"</DownstreamJoint>
        <UpstreamJoint>"RightAnkle"</UpstreamJoint>
     <Plane>"Frontal"</Plane>
     <AlphaAngle>"0"</AlphaAngle>
     <MaxAngleDeviation> "15" </MaxAngleDeviation>
    </Configuration>
    <Configuration>
        <Type>"JointAngle"</Type>
        <CenterJoint>"RightKnee"</CenterJoint>
        <DownstreamJoint>"HipCenter"</DownstreamJoint>
        <UpstreamJoint>"RightAnkle"</UpstreamJoint>
        <Angle>"180"</Angle>
        <MaxAngleDeviation>"15"</MaxAngleDeviation>
    </Configuration>
  </InvarianceRule>
</ExerciseRules>
```

Listing 8. **The rules for the hip-abduction exercise.**

the initial configuration. This leads to a dynamic rule with two reference configurations, one for the initial pose and the other for the final pose where the abducting leg reaches the out-most position, as shown in Figure 5. Furthermore, there is an invariance rule to dictate that the abducting leg must move within the frontal plane. The rules for the hip-abduction exercise is given in Listing 8.

C. SIT TO STAND

The sit-to-stand exercise requires a user to first sit, and then lean forward to stand up, as shown in Figure 6. When in the sitting position, the user's two feet must be placed evenly on the floor, and the torso is straight up so that the angle formed between the torso and the legs is approximately 90 degrees. While the user is standing up, the feet should remain at the initial place.

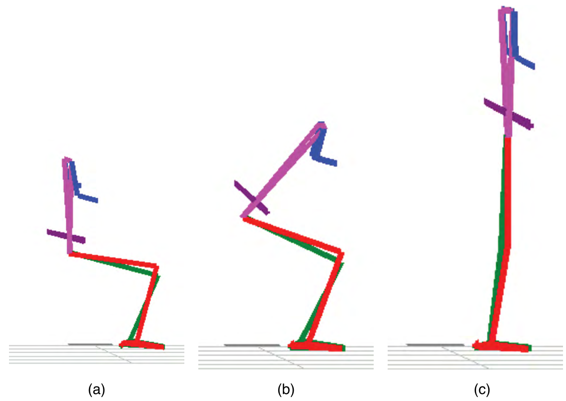


FIGURE 6. **The three reference configurations for the sit-to-stand exercise from the sagittal view. (a) The initial pose; (b) intermediate pose; and (c) the final pose.**

To make multiple repetition of the exercise, a user would perform the mirrored activity, stand to sit, after the sit-to-stand activity. Unlike the bowling and the hip abduction exercises, where the end pose of the first activity is not stable, the sit-to-stand activity ends with a stable pose because he/she may choose to remain in the standing pose for some time before proceeding to the stand-to-sit activity. As such, it is more robust to track the sit-to-stand, and the stand-to-sit activities separately using two finite state machines.

Here we only define the rules only for the sit-to-stand activity, as shown in Listing 9. The rules for the stand-to-sit activity is rather similar (the only difference is that the reference configurations in the dynamic rule are in reverse order). One dynamic rule and one static rule are used to define the sit-to-stand exercise. The dynamic rule defines the hip angle movement. The sit-to-stand activity has the two monotonic segments using the hip angle as the variable. Hence, there are three reference configurations in the dynamic rule, as shown in Figure 6, for the initial, intermediate, and final pose with the hip angle at 90 degrees, 60 degrees, and 180 degrees, respectively. The static rule describes the rule about the foot placement, *i.e.,* they must be aligned in parallel to the frontal plane.

VI. SYSTEM DESIGN AND IMPLEMENTATION

In this section, we present the design and implementation of an avatar-based system [23], [24] for rehabilitation exercises guidance and realtime assessment. The system provides guidance to its users by demonstrating in a virtual 3D environment the proper way of practicing an exercise via an avatar on the left side of the screen. The demonstration is enabled by replaying in a loop pre-recorded motion data. The data can be recorded from a user under the supervision of a physical therapist, or directly recorded from a physical therapist doing the exercise. On the right side of the screen, another avatar

```
<ExerciseRules Name="Sit to Stand">
  <DynamicRule>
    <Configuration>
         <Type>"JointAngle"</Type>
         <CenterJoint>"HipCenter"</CenterJoint>
         <DownstreamJoint>"ShoulderCenter"
              </DownstreamJoint>
         <UpstreamJoint>"Left Knee"</UpstreamJoint>
         <Angle>"90"</Angle>
         <MaxAngleDeviation> "20" </MaxAngleDeviation>
    </Configuration>
    <Configuration>
         <Type>"JointAngle"</Type>
         <CenterJoint>"HipCenter"</CenterJoint>
         <DownstreamJoint>"ShoulderCenter"
              </DownstreamJoint>
         <UpstreamJoint>"LeftKnee"</UpstreamJoint>
         <Angle>"60"</Angle>
         <MaxAngleDeviation> "20" </MaxAngleDeviation>
    </Configuration>
    <Configuration>
        <Type>"JointAngle"</Type>
        <CenterJoint>"HipCenter"</CenterJoint>
        <DownstreamJoint>"ShoulderCenter"
             </DownstreamJoint>
        <UpstreamJoint>"LeftKnee"</UpstreamJoint>
        <Angle>"180"</Angle>
         <MaxAngleDeviation> "10" </MaxAngleDeviation>
    </Configuration>
  </DynamicRule>
  <StaticRule>
    <Configuration>
        <Type>"BoneOrientation"</Type>
        <DownstreamJoint>"LeftAnkle"</DownstreamJoint>
        <UpstreamJoint>"RightAnkle"</UpstreamJoint>
        <Plane>"Frontal"</Plane>
        <AlphaAngle>"0"</AlphaAngle>
        <MaxAngleDeviation> "20" </MaxAngleDeviation>
    </Configuration>
  </StaticRule>
<ExerciseRules>
```

Listing 9. The rules for the sit-to-stand exercise.

shows the actual user movement in realtime. Furthermore, the system uses the rules defined for the current exercise to assess the user performance in realtime. The quantity and the quality of the exercise performed are presented to the user in the form of visual aids in realtime. The system also logs information such as the number of correct repetitions and raw motion data (*i.e.,* joint positions) for offline analysis. If desirable, the log data can be furnished to the user's physical therapist to review.

The system is designed to satisfy the following requirement:

- The system should provide a three-dimensional visual guide on how to perform a rehabilitation exercise prescribed by a therapist. The user should be able to view the movement from different angles and in slow motion so that he/she could learn the correct movement.
- The system should not display images of the demonstrator or the user so that the user feels more comfortable practicing the prescribed exercises. This may seem to be counter-intuitive. However, numerous studies have shown that users typically prefer not to see their own images when doing rehabilitation exercises.
- The system should provide realtime feedback to the user while he/she is practicing. In particular, the system should catch a wrong movement as soon as it is performed to minimize the risk of injuries. Furthermore, the feedback should be presented to the user at least visually with intuitive cues.
- For review by the therapist, the movements of a user should be recorded as well as higher-level meta data regarding the performance of the exercises practiced by a user at home. This is important to facilitate patient accountability and tele-medicine.

To satisfy the first two requirements, we choose to use the Unity 3D game development framework (https://unity3d.com/). While the Microsoft software development kit (SDK) and virtually every third party toolkit offers application programming interfaces (APIs) and sample applications for skeletal data rendering, the display is in 2-dimensional only with the joints connected by artificial bones. Such an approach is not satisfactory because it fails to show the movements in the sagittal plane. Even though overlapping color images of the demonstrator or the user with the skeleton images could result in a better depth view, the second requirement (about not showing the user images) rules out this approach.

Indeed, with the Unity framework, both the coach and the user are represented by three-dimensional avatars. Furthermore, the movements of the coach and the user can be watched frame-by-frame in a 360-degree view. This help the user to learn how to perform a newly prescribed exercise, and facilitates the therapist to examine the user's practicing quality in great detail.

To satisfy the third requirement, the system implements a motion assessment engine that is capable of (1) parsing the rules for each rehabilitation exercise and creating internal data structures accordingly, and (2) examining each incoming skeleton frame according to the rules and assess the quantity and quality in realtime. Our rule-based framework (*i.e.,* the rule specification as well as the motion assessment mechanisms) makes this task possible. Furthermore, in addition to the avatars, we added target positions for each exercise for key joints. To facilitate this extension, we extend the rules by adding a new element <TargetJoint>. If a joint in a final reference configuration of a dynamic rule is included in the <TargetJoint> element, a sphere is inserted at the target position. This sphere is used as the visual cue in the following ways: (1) it shows the number of correct repetition count; (2) the sphere is colored green as long as the user is performing the repetition correctly, and it turns yellow as soon as the user violates one of the rules. If the latter happens, the current repetition is not counted, and an additional text indicating the nature of the error is displayed on the screen.

To satisfy the last requirement, the joint positions in Cartesian coordinates as well as the segment orientations in quaternion of the user for every frame are captured and logged to files. Furthermore, the assessment results and the repetition counts are also logged.

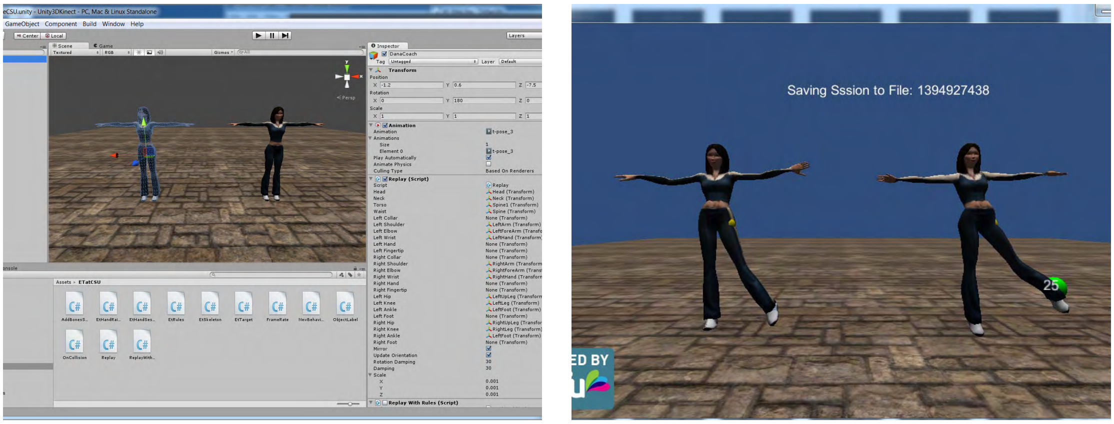


FIGURE 7. Our system: (a) in the project view, and (b) in the game view.

We implemented the system as a Unity project [25], as shown in Figure 7. We choose to use the ZigFu plugin (http://developkinect.com/groups/zigfu; the official website is not longer accessible) to access the Kinect runtime. Because the ZigFu plugin supports the Microsoft Kinect SDK as well as OpenNI, both the Kinect sensor and any of the OpenNI-compatible depth sensors (such as Asus Xtion Pro Live) can be used with our system. The C# programming language is used to script the avatars and all dynamic content of the virtual reality environment.

As shown in the project view (Figure 7(a)). The main assets are two avatars provided by the ZigFu plugin:

- CoachAvatar. This avatar is placed on the left side of the scene. A C# script named Replay.cs is used to controls the movement of the CoachAvatar using the motion data collected previously. This avatar serves as the virtual coach for the user to demonstrate correct movement.
- SubjectAvatar. This avatar is placed on the right side of the scene. Another script named EtSkeleton.cs is used to control the avatar using the captured motion data while a user is performing the rehabilitation exercise in realtime.

The scene also contains several statically allocated components, include the floor, directional light, main camera, status display, and an invisible game object used to attach the ZigFu runtime scripts for motion data capture. The scene also contains the visual cuing objects created dynamically according to the correctness rules.

VII. HUMAN SUBJECT STUDY

The avatar-based system was used in a human subject study with eight participants. The goal of the human subject study was to validate if the system is helpful in improving the quality of exercises performed unsupervised by a physical therapist. However, we will report the findings in a separate article regarding the validation of this hypothesis. In this paper, we present evidence that the system can be used to provide realtime assessment and feedback to users of different profiles. In this study, four males and four females participated. Their height profiles are shown in Figure 8. As can be seen, the participants range from as short as 1.45 meters to as tall as 1.86 meters. We report five repetitions of each participant for the first two exercises, and a single repetition for the sit-to-stand exercise. We show that with the right parameters being used in the correctness rules for the three exercises we have experimented, the assessment can be done for all participants.

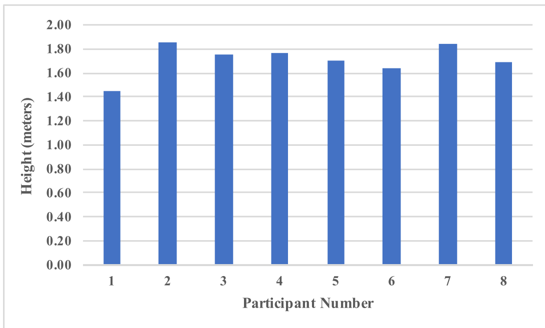


FIGURE 8. Height profiles for eight subjects.

A. RESULT FOR BOWLING

The result for the bowling exercise is shown in Figure 9. To fit all results within the range of 0-180 degrees, the bowling arm orientation angle, *beta*, is transformed to be 360 − *beta*. Hence, the initial pose angle would be close to 90 degrees instead of 270 degrees, and the final pose angle would remain to be around 180 degrees. As can be seen from Figure 9, the final poses for all eight subjects are within 20 degrees from the ideal angle (*i.e.,* 180 degrees), and the initial pose angles also fluctuates between 90-110 degrees. This observation suggests that we should use 20 degrees as the tolerance value for the *beta* angle. All five repetitions done by all eight subjects were considered correct by the physical therapists participating this study. The invariance rule on the arm movement with respect to the sagittal plane is assessed by the angle between the arm vector and the sagittal plane. In Figure 9,

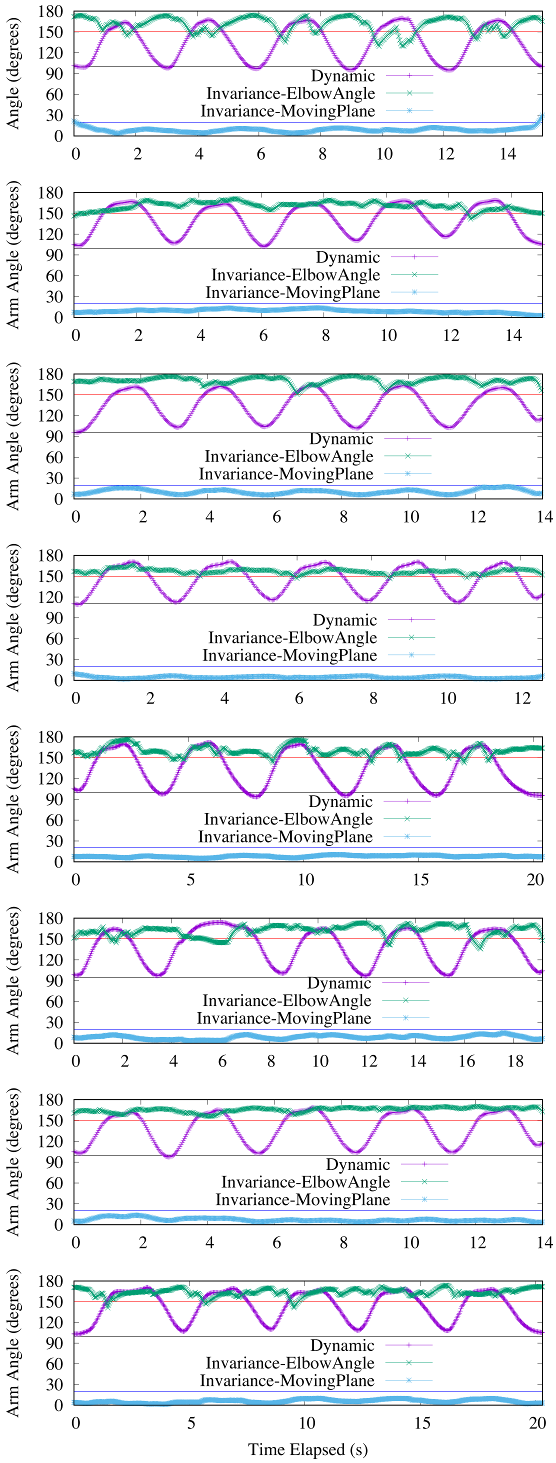


FIGURE 9. Raw data for five repetitions of the bowling exercise recorded for eight subjects.

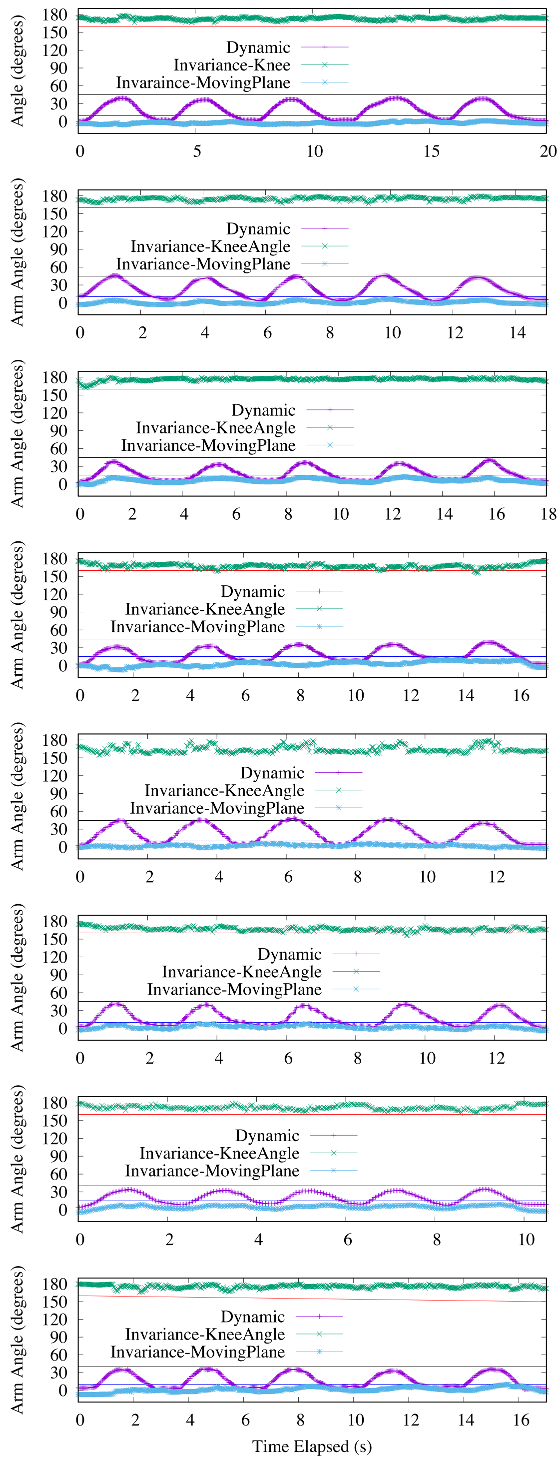


FIGURE 10. Raw data for five repetitions of the hip-abduction exercise recorded for eight subjects.

this curve is labeled as "Invariance-MovingPlane". As can be seen, this angles varies between 0 and 20 degrees. This suggests the tolerance value should be set to 20 degrees.

However, the bowling arm elbow angle (labeled as "Invariance-ElbowEngle" in the figure) (*i.e.,* for the invariance rule that the arm should remain straight during the movement) varies significantly beyond a 30-degree range for some subjects (*i.e.,* subjects 1, 6, and 8). This observation means that even with a generous tolerance value of 30 degrees for the elbow angle rule, the system would label some of the repetitions done by these three subjects as wrong. One solution is to this problem is to set personalized tolerance values for different subjects to fit their profiles.

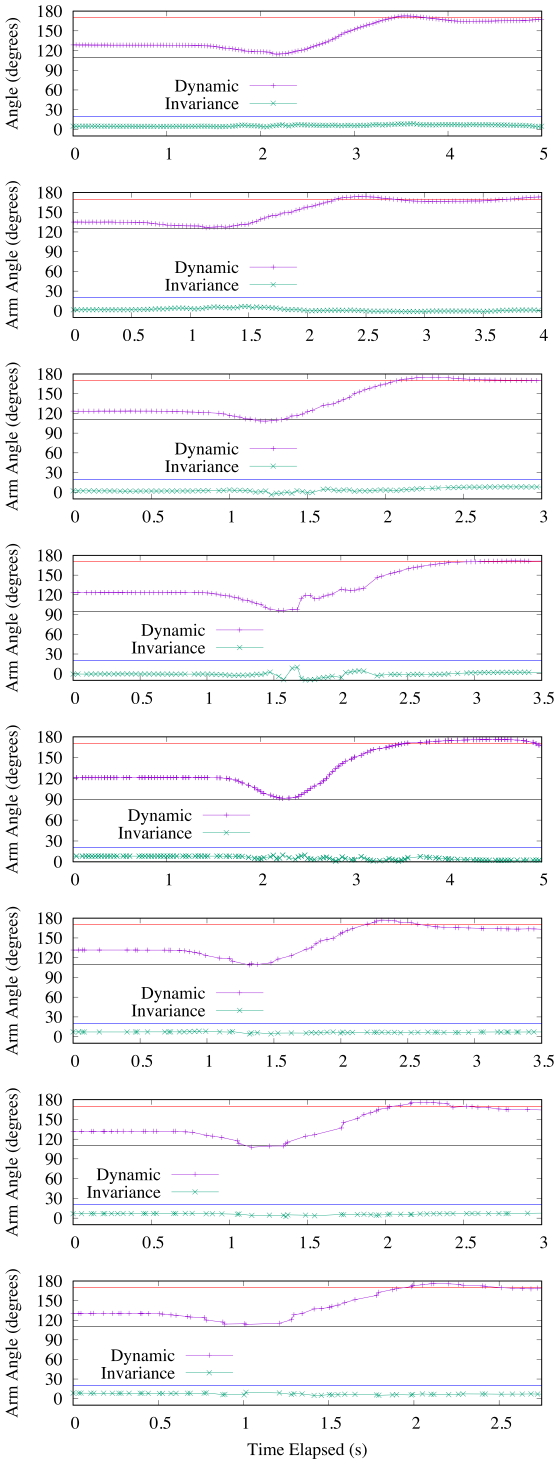


FIGURE 11. Raw data for one repetition of the sit-to-stand exercise recorded for eight subjects.

B. RESULT FOR HIP ABDUCTION

The result for the hip abduction exercise is shown in Figure 10. As can be seen, the variations between these eight subjects are much less prominent than those for the bowling exercise. The abducting leg initial pose hip angle varies between 0 and 15 degrees. The final pose hip angle varies between 40 and 45 degrees. The knee angles (for the invariance rule on keeping the abducting leg straight) are within the range of 160-180 degrees. The angle between of the abducting leg and the frontal plane (for the invariance rule on moving within the frontal plane) varies between -10 to 15 degrees. These suggest that the tolerance value for the initial pose should be set to 15 degrees, the tolerance value for the final pose can be set as small as 5 degrees, the tolerance value for the knee angle should be set to 20 degrees, and the tolerance value for the abducting leg orientation *beta* angle should be set to 15 degrees.

C. RESULT FOR SIT TO STAND

The result for the sit-to-stand exercise is shown in Figure 11. This experiment revealed that it is a challenge to use the Kinect-based system to monitor the sit-to-stand exercise because the initial pose hip angle and the intermediate pose hip angle are significantly different from the expected values. For the initial value, we expect 90 degrees, and for the intermediate pose, we expect about 60 degrees. The actual result shows that the initial angle varies between 120 to 130 degrees, and the intermediate pose hip angles vary in the range of 90 to 110 degrees. A detailed analysis of the Kinect data compared with a multi-camera motion tracking system confirms that Kinect has systematic error in determining the hip center and spine joints [26]. However, because the error is systematic, we can workaround the issue by shifting the expected angle values, from 90 to 120 degrees for the initial pose, and from 60 to 90 degrees for the intermediate pose. The final pose hip angle is within 20 degrees from the expected 180-degree value. With this adjustment, the Kinect-system can in fact reliably be used to assess the performance of sit-to-stand exercise.

VIII. CONCLUSION

In this article, we presented a rule-based framework for defining key requirements on properly executing rehabilitation exercises. We also elaborated the design and implementation of an avatar-based guidance and monitoring system that incorporates the rule-based framework, and reported experimental results with eight healthy human subjects. Our system is intended to be used at home for a user to carry out the prescribed rehabilitation exercises without direct in-person supervision of physical therapists. We show that as long as the parameters for the exercises, including the expected values and tolerance values, are configured properly, our system can be used reliably to give users realtime guidance and feedback. The use of inexpensive Microsoft Kinect sensor makes the system a low-cost, and possibly more engaging tool for users to practice at home. In the future, the system can be extended to include cloud services to enable tele-monitoring [27]–[30] of the user performance by physical therapists. Finally, it is worth noting that the rule-based framework is not only useful for specifying correctness requirements for rehabilitation exercises, but it also can be used to specify other human activities, such as using proper body mechanics when performing pulling and lifting activities [31]. The framework proves to be valuable in building systems that help promote healthier workplaces [31]–[35] and healthier life styles [36], [37].

REFERENCES

[1] J. A. Kleim and T. A. Jones, "Principles of experience-dependent neural plasticity: Implications for rehabilitation after brain damage," *J. Speech, Lang., Hearing Res.*, vol. 51, no. 1, pp. S225–S239, 2008.

[2] S. McCallum, "Gamification and serious games for personalized health," *Stud Health Technol. Inf.*, vol. 177, pp. 85–96, Jan. 2012.

[3] E. Velloso, A. Bulling, and H. Gellersen, "MotionMA: Motion modelling and analysis by demonstration," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2013, pp. 1309–1318.

[4] R. F. Escamilla, K. Yamashiro, L. Paulos, and J. R. Andrews, "Shoulder muscle activity and function in common shoulder rehabilitation exercises," *Sports Med.*, vol. 39, no. 8, pp. 663–685, 2009.

[5] D. Tino and C. Hillis, "The full can exercise as the recommended exercise for strengthening the supraspinatus while minimizing impingement," *Strength Conditioning J.*, vol. 32, no. 5, pp. 33–35, 2010.

[6] X. Luo, J. Deng, J. Liu, W. Wang, X. Ban, and J.-H. Wang, "A quantized kernel least mean square scheme with entropy-guided learning for intelligent data analysis," *China Commun.*, vol. 14, no. 7, pp. 1–10, 2017.

[7] X. Luo, J. Deng, W. Wang, J.-H. Wang, and W. Zhao, "A quantized kernel learning algorithm using a minimum kernel risk-sensitive loss criterion and bilateral gradient technique," *Entropy*, vol. 19, no. 7, p. 365, 2017.

[8] X. Luo *et al.*, "Towards enhancing stacked extreme learning machine with sparse autoencoder by correntropy," *J. Franklin Inst.*, to be published. [Online]. Available: https://doi.org/10.1016/j.jfranklin.2017.08.014, doi:10.1016/j.jfranklin.2017.08.014.

[9] X. Luo, Y. Lv, M. Zhou, W. Wang, and W. Zhao, "A laguerre neural network-based adp learning scheme with its application to tracking control in the Internet of Things," *Pers. Ubiquitous Comput.*, vol. 20, no. 3, pp. 361–372, 2016.

[10] X. Luo, D. Zhang, L. T. Yang, J. Liu, X. Chang, and H. Ning, "A kernel machine-based secure data sensing and fusion scheme in wireless sensor networks for the cyber-physical systems," *Future Generat. Comput. Syst.*, vol. 61, pp. 85–96, Aug. 2016.

[11] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," *Commun. ACM*, vol. 56, no. 1, pp. 116–124, Jan. 2013.

[12] C.-J. Su, "Personal rehabilitation exercise assistant with Kinect and dynamic time warping," *Int. J. Inf. Edu. Technol.*, vol. 3, no. 4, pp. 448–454, 2013.

[13] L. Zhang, J.-C. Hsieh, and J. Wang, "A Kinect-based golf swing classification system using HMM and neuro-fuzzy," in *Proc. Int. Conf. Comput. Sci. Inf. Process. (CSIP)*, Aug. 2012, pp. 1163–1166.

[14] B. C. Bedregal, A. C. Costa, and G. P. Dimuro, "IFIP international federation for information processing," in *Artificial Intelligence in Theory and Practice*, vol. 217, M. Bramer, Ed. Boston, MA, USA: Springer, 2006, pp. 285–294.

[15] T. Hachaj and M. R. Ogiela, "Rule-based approach to recognizing human body poses and gestures in real time," *Multimedia Syst.*, vol. 20, no. 1, pp. 81–99, 2014.

[16] R. A. Clark *et al.*, "Validity of the Microsoft Kinect for assessment of postural control," *Gait Posture*, vol. 36, no. 3, pp. 372–377, 2012.

[17] R. A. Clark, Y.-H. Pua, A. L. Bryant, and M. A. Hunt, "Validity of the Microsoft Kinect for providing lateral trunk lean feedback during gait retraining," *Gait Posture*, vol. 38, no. 4, pp. 1064–1066, 2013.

[18] A. Bo, M. Hayashibe, and P. Poignet, "Joint angle estimation in rehabilitation with inertial sensors and its integration with Kinect," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2011, pp. 3479–3483.

[19] E. Akdoğan, E. Taçgın, and M. A. Adli, "Knee rehabilitation using an intelligent robotic system," *J. Intell. Manuf.*, vol. 20, no. 2, pp. 195–202, 2009.

[20] Q. Wang, P. Turaga, G. Coleman, and T. Ingalls, "SomaTech: An exploratory interface for altering movement habits," in *Proc. Extended Abstracts Hum. Factors Comput. Syst. CHI*, 2014, pp. 1765–1770.

[21] W. Zhao, R. Lun, D. D. Espy, and M. Reinthal, "Rule based realtime motion assessment for rehabilitation exercises," in *Proc. IEEE Symp. Comput. Intell. Healthcare e-Health*, Dec. 2014, pp. 133–140.

[22] W. Zhao, R. Lun, D. D. Espy, and M. A. Reinthal, "Realtime motion assessment for rehabilitation exercises: Integration of kinematic modeling with fuzzy inference," *J. Artif. Intell. Soft Comput. Res.*, vol. 4, no. 4, pp. 267–285, 2014.

[23] W. Zhao, "A concise tutorial on human motion tracking and recognition with microsoft kinect," *Sci. China Inf. Sci.*, vol. 59, no. 9, p. 93101, 2016.

[24] R. Lun and W. Zhao, "A survey of applications and human motion recognition with Microsoft Kinect," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 29, no. 5, p. 1555008, 2015.

[25] W. Zhao, H. Feng, R. Lun, D. D. Espy, and M. Reinthal, "A Kinect-based rehabilitation exercise monitoring and guidance system," in *Proc. 5th IEEE Int. Conf. Softw. Eng. Service Sci.*, Jun. 2014, pp. 762–765.

[26] W. Zhao, D. D. Espy, M. Reinthal, and H. Feng, "A feasibility study of using a single Kinect sensor for rehabilitation exercises monitoring: A rule based approach," in *Proc. IEEE Symp. Comput. Intell. Healthcare e-Health*, 2014, pp. 1–8.

[27] T. Qiu, A. Zhao, F. Xia, W. Si, and D. O. Wu, "ROSE: Robustness strategy for scale-free wireless sensor networks," *IEEE/ACM Trans. Netw.*, to be published. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7956295&isnumber=4359146, doi: 10.1109/TNET.2017.2713530.

[28] T. Qiu, R. Qiao, and D. Wu, "EABS: An event-aware backpressure scheduling scheme for emergency Internet of Things," *IEEE Trans. Mobile Comput.*, to be published. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7922596&isnumber=4358975, doi: 10.1109/TMC.2017.2702670.

[29] T. Qiu, K. Zheng, H. Song, M. Han, and B. Kantarci, "A local-optimization emergency scheduling scheme with self-recovery for smart grid," *IEEE Trans. Ind. Informat.*, to be published. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7951030&isnumber=4389054, doi: 10.1109/TII.2017.2715844.

[30] T. Qiu, Y. Zhang, D. Qiao, X. Zhang, M. L. Wymore, and A. K. Sangaiah, "A robust time synchronization scheme for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, to be published. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8008773&isnumber=4389054, doi: 10.1109/TII.2017.2738842.

[31] W. Zhao *et al.*, "A human-centered activity tracking system: Toward a healthier workplace," *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 3, pp. 343–355, Jun. 2017.

[32] W. Zhao, D. D. Espy, M. Reinthal, B. Ekelman, G. Goodman, and J. Niederriter, "Privacy-aware human motion tracking with realtime haptic feedback," in *Proc. IEEE Int. Conf. Mobile Services*, Jun. 2015, pp. 446–453.

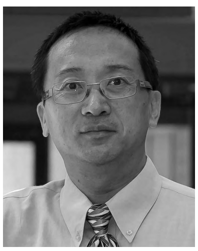
[33] W. Zhao, R. Lun, D. D. Espy, and M. A. Reinthal, "A privacy-aware Kinect-based system for healthcare professionals," in *Proc. IEEE Int. Conf. Electro-Inf. Technol.*, Grand Forks, ND, USA, May 2016, pp. 205–210.

[34] W. Zhao *et al.*, "LiftingDoneRight: A privacy-aware human motion tracking system for healthcare professionals," *Int. J. Handheld Comput. Res.*, vol. 7, no. 3, pp. 1–15, 2016.

[35] W. Zhao *et al.*, "A privacy-aware compliance tracking system for skilled nursing facilities," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Banff, AB, Canada, Oct. 2017, pp. 3568–3573.

[36] W. Zhao and R. Lun, "A Kinect-based system for promoting healthier living at home," in *Proc. IEEE Int. Conf. Syst., Man, (SMC)*, Oct. 2016, pp. 000258–000263.

[37] R. Lun, C. Gordon, and W. Zhao, "Tracking the activities of daily lives: An integrated approach," in *Proc. Future Technol. Conf. (FTC)*, 2016, pp. 466–475.



WENBING ZHAO (S'99–M'02–SM'14) received the Ph.D. degree in electrical and computer engineering from the University of California, Santa Barbara, CA, USA, in 2002. He is currently a Professor and the Director of the M.Sc. degree in electrical engineering program with the Department of Electrical Engineering and Computer Science, Cleveland State University. His current research interests include dependable distributed systems, smart, and connected health. He has over 150 academic publications and has received best paper awards at several international conferences. He has served on the organizing and technical committees of numerous conferences. He is an Associate Editor for the IEEE Access and an Academic Editor for PeerJ Computer Science, and a member of the editorial boards of several international journals.

M. ANN REINTHAL received the Ph.D. degree in neuroscience and practiced physical therapy in a variety of settings in a faculty position at Cleveland State University in 1997, and where she is currently an Associate Professor. Her research interests are in the areas of motor control and learning. She is especially interested in integrating the use of technology into rehabilitation for neurologically impaired populations. She has done previous research using various commercial video gaming technologies to improve upper extremity function and balance.

DEBORAH D. ESPY received the Ph.D. in human movement science from the University of Illinois at Chicago. She is currently an Assistant Professor in the Physical Therapy Program with Cleveland State University. Her research interests are in the areas of motor learning, and postural control and falls. Her current lines of research are in the use of video gaming as a therapeutic exercise and balance training modality; in novel uses of sensor technology in motion feedback, and in balance training for fall prevention.

XIONG LUO received the Ph.D. degree from Central South University, China, in 2004. He is currently a Professor with the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His current research interests include machine learning, cloud computing, and computational intelligence. He has published extensively in his areas of interest in journals, such as the IEEE Access, *Future Generation Computer Systems*, and *Personal and Ubiquitous Computing*.