Exercise Recognition for Kinect-based Telerehabilitation*

D. Antón; A. Goñi; A. Illarramendi
Department of Computer Languages and Systems, University of the Basque Country UPV/EHU, Donostia-San Sebastián, Spain

Keywords
Telerehabilitation, telemedicine, exercise recognition, Kinect-based motion tracking

Summary
Background: An aging population and people’s higher survival to diseases and traumas that leave physical consequences are challenging aspects in the context of an efficient health management. This is why telerehabilitation systems are being developed, to allow monitoring and support of physiotherapy sessions at home, which could reduce healthcare costs while also improving the quality of life of the users.

Objectives: Our goal is the development of a Kinect-based algorithm that provides a very accurate real-time monitoring of physical rehabilitation exercises and that also provides a friendly interface oriented both to users and physiotherapists.

Methods: The two main constituents of our algorithm are the posture classification method and the exercises recognition method. The exercises consist of series of movements. Each movement is composed of an initial posture, a final posture and the angular trajectories of the limbs involved in the movement. The algorithm was designed and tested with datasets of real movements performed by volunteers. We also explain in the paper how we obtained the optimal values for the trade-off values for posture and trajectory recognition.

Results: Two relevant aspects of the algorithm were evaluated in our tests, classification accuracy and real-time data processing. We achieved 91.9% accuracy in posture classification and 93.75% accuracy in trajectory recognition. We also checked whether the algorithm was able to process the data in real-time. We found that our algorithm could process more than 20,000 postures per second and all the required trajectory data-series in real-time, which in practice guarantees no perceptible delays. Later on, we carried out two clinical trials with real patients that suffered shoulder disorders. We obtained an exercise monitoring accuracy of 95.16%.

Conclusions: We present an exercise recognition algorithm that handles the data provided by Kinect efficiently. The algorithm has been validated in a real scenario where we have verified its suitability. Moreover, we have received a positive feedback from both users and the physiotherapists who took part in the tests.

1. Introduction
An aging population and people’s higher survival to diseases and traumas that leave physical sequels are challenging aspects in the context of an efficient health management. Telemonitoring technologies have been proposed as a solution to reduce hospital overloads, and using such technologies data can be accessed remotely by healthcare professionals through the Internet and mobile devices [1]. In the area of physiotherapy, telerehabilitation systems that support physiotherapy sessions at home could help reduce healthcare costs while also improving the quality of life of the users that need rehabilitation. Cost containment in health care while trying to maintain access to quality services has become essential in the last years, as we face an aging population [2].

Telerehabilitation should not be seen as a technology in itself, but as the use of new technologies to improve and optimize both rehabilitation services and patient outcome with the idea of reinforcing traditional rehabilitation [3]. Several studies have demonstrated that virtual interaction can be as effective as traditional treatments [4, 5]. Furthermore, the use of telerehabilitation systems with motion capture has been shown to increase the intensity of rehabilitation and enhance user experience [4, 6].

The core technology of our telerehabilitation system is Kinect, an innovative motion capture device developed by Microsoft [7] and PrimeSense. In the specialized literature we can find works that suggest that Kinect can validly assess kinematic strategies of postural control such as [8]. There are also works [9, 10] that suggest that the validity of Kinect posture estimation is comparable to more established techniques for posture estimation from 3D motion capture data. Kinect allowed us to create an innovative telerehabilitation system that can automatically evaluate user’s exercises by recognizing user’s movements.

The focus of this paper is the algorithm that recognizes and evaluates the therapeutic exercises. We present how exercises are described and how they are recognized. We also introduce some performance results that show the good behavior of the proposed algorithm.
This paper is organized as follows: In section 2 we describe some previous works done in this field. Next, in section 3 we explain the main features of the methods that constitute our algorithm. In section 4 we present the datasets used for the experiments and some initial considerations related to them, and in section 5 the results obtained. Finally, in sections 6 and 7 we present the discussion and some conclusions respectively.

2. Background

Telerehabilitation systems can be found both in an academic setting as in a commercial environment. If we analyze their evolution we can observe that some of them make use of wearable devices (e.g. [11, 12]). In [11] Llorens et al. present Bio-track, a system for task-oriented games that evaluates whether people with cognitive impairment can reach some predefined locations. To that end, the system makes use of markers attached to the user's body and infrared cameras. In [12] the authors use smartphone's build-in inertial sensors to monitor exercise execution and to provide acoustic feedback on exercise performance and execution errors. However, a trend is seen nowadays for the use of low-cost non-intrusive tracking devices such as Nintendo Wii Remote or Kinect in the telerehabilitation systems. In [13] the authors describe a telerehabilitation system, based on Nintendo Wii Remote, which uses an accelerometer to record the user's movements in 3D. The system focuses on rehabilitation exercises of upper limbs. Among the proposals that use Kinect two groups can be distinguished: proposals that make use of Kinect for Xbox; and those that make use of Kinect for Windows. Among the works of the first group we can mention [14–16]. In [14] the authors present a prototype of a game-based telerehabilitation system with Kinect that they have developed. However, their main goal is to prove the adequacy of using Kinect for telerehabilitation therapies and so they do not show technical details about the recognition method. In [15] Kinerehab is presented, an occupational therapy system based on Kinect, where users can perform three different exercises: lift arms front, lift arms sides and lift arms up. Finally, in [16] they present 21 game concept prototypes which receive and process data sent by Kinect but the authors do not deal with the evaluation. Concerning the works that use Kinect for Windows we can find, on the one hand, commercial products such as [17, 18] which do not show many technical details concerning their internal behavior. On the other hand, there are research proposals that focus on different pathologies, such as [19] which focuses on patients with traumatic brain injury, [20, 21] which are oriented to upper limb rehabilitation, and [22] which is focused on full body gait analysis. Moreover, we want to mention the system presented in [23], which explores the combined use of inertial sensors and Kinect. They made an evaluation of different exercises (shoulder abduction and adduction, squat and sit to stand), but their goal was more aimed at performing online calibration of sensor errors than the evaluation of the exercises.

Our proposal advocates for the use of Kinect, and in this paper, we focus on the core exercise recognition algorithm of our telerehabilitation system. Next, we mention the main characteristics that distinguish it from other proposals from three different perspectives.

1) From the users' point of view, the algorithm provides visual and acoustic feedback about the exercises performed so that users can see, through avatars, how they are doing the exercises and how the therapist performed them (Figure 1). They can also see the number of series that remain to perform and also the number of remaining repetitions for the actual exercise respectively. Moreover, when the user reaches the final posture, an acoustic signal is provided together with information.
about the speed of execution (adequate, too fast or too slow).

2) From the therapists’ point of view, the algorithm allows them to define exercises for the users by a) using exercises already stored in a library, b) combining those stored exercises, or/and c) defining new customized exercises simply by recording them in front of Kinect. The way exercises are recorded and stored facilitates their exchange among different therapists. Furthermore, therapists can reproduce in their computers the sessions that users have made at their homes. The general idea is to mimic their usual way of working (Figure 2).

3) From the technical point of view, we want to mention three main features, a) an efficient real-time execution, so users get on-line feedback; b) a good accuracy when recognizing exercises; and c) flexibility when adapting itself to the user’s body movement limitations at each moment. The algorithm not only considers final snapshots of the exercises performed but also intermediate snapshots during executions, which means that the goal is not only to identify the final posture but also how well the user gets to it.

Finally, the algorithm has been validated with data taken from volunteers. Those data are available in http://bdi.si.ehu.es/bdi/members/david-anton/research-resources/. Moreover, it has also been tested with real users. In both cases, the algorithm provides good accuracy at recognizing exercises.

3. Methods

In this section, we first show the descriptor that models body postures and the method used to classify those postures. Then we present the main features of the exercise recognition method.

3.1 The Descriptor of Postures

Kinect is a visual tracking system without markers that allows users to control and interact with applications through an interface that can recognize gestures, voice commands and objects. Kinect provides a skeleton structure in which each node is a joint in the body. The skeleton contains a total of 20 joints described by 3D points. These points are referenced in a coordinate system (axes X, Y and Z) whose origin is at the center of the plane parallel to the captured image and intersecting with the Kinect camera. The coordinates obtained from Kinect are translated to another coordinate system whose origin is at the hip center of the user so that relative position between the camera and the user does not influence the exercise recognition. Those translated coordinates are used to calculate the following three types of measurements: 1) Relative positions of some parts of the body in the Z axis. A volume around the user is defined by two values, a minimum and a maximum distance in the Z axis, and two binary features for each joint are generated: one that takes the value 1 or 0 depending on whether the Z coordinate of a joint is above the minimum, and the other one that takes the value 1 or 0 depending on whether the Z coordinate of a joint is below the maximum. 2) Angles between joints. They are the angles between the lines formed by two joints, relative to the origin of coordinates located at the first one of them and 3) Angles between limbs. They are the angles between two limbs connected by a joint.

With these values we define a posture descriptor that reduces significantly the dimensionality of the data. We obtain a simplified representation of a body posture that still encompasses sufficient information for the recognition process as we show in Section 5. The posture descriptor has a total of 30 features, divided in two distinct parts (Table 1), 18 binary features (from 1 to 18) that give information about the relative position in 3D of some joints (neck, hands, shoulders, knees and feet) and 12 features that represent the angles formed by the different parts of the body projected in the frontal plane (XY) (from 20 to 24 and from 26 to 30) and in the lateral plane (XZ) (19 and 25).

3.2. Posture Classification Method

Once a posture is captured and its corresponding descriptor is created, the next step is to classify it. Classification is made by comparing the captured descriptor with previously annotated posture descriptors. To compare two posture descriptors $D_i$ and $D_j$, a similarity measurement, $\text{dist}(D_i, D_j)$, based on the distance between them is used:

$$\text{dist}(D_i, D_j) = \text{angDiff}(D_i, D_j) \times (1 + \text{binDist}(D_i, D_j)) \quad (1)$$

As mentioned before, the descriptor is composed of two parts: on the one hand, a set of 18 binary features and, on the other hand, 12 angular measurements of body members. The two parts of the descriptor (binDist($D_i$, $D_j$) and angDiff($D_i$, $D_j$)) are evaluated independently, by using formulas...
Exercise Recognition for Kinect-based Telerehabilitation

D. Antón et al.: Exercise Recognition for Kinect-based... 2/2015 © Schattauer 2015

148 D. Antón et al.: Exercise Recognition for Kinect-based... posture descriptor to classify and the annotated posture descriptor is in fact assigned.

When different posture classes could be assigned, the one with the smallest distance between the posture descriptor to classify and the annotated posture descriptor is in fact assigned.

It is quite obvious that the lower the threshold value $pth_0$, the greater the similarity between the compared posture descriptors must be. In the event that $pth_0$ were 0, then the user must perform a posture that is exactly the same as one that has been previously recorded in order to be classified as that. However, it must be noticed that there are different descriptors annotated with the same posture class. Therefore, using a threshold $pth_0 = 0$ may be not appropriate when the performed posture descriptor is not exactly equal to any of the recorded ones but is definitely of that posture. On the contrary, greater values for the threshold would make a posture descriptor be misclassified. In section 5.1.1 we show which is the optimal value obtained for this trade-off value that is $pth_0$.

3.3 Exercise Recognition Method

In rehabilitation therapies, exercises usually consist of series of movements. Each movement is composed of an initial posture, a final posture and the angular trajectories of the limbs involved in the movement (the relevant limbs). Both the initial posture and the final posture of a movement are identified with their respective descriptors. The movement between the initial and final posture is represented by sequences of angular values taken from the limbs that are in a different position from one posture to another (it is assumed that the limbs whose positions are equal in the initial and in the final postures do not move during the transition). Complex exercises are defined as a combination of basic movements, creating a sequence of movements where the final posture of a movement matches the initial posture of the next one.

3.3.1 Identification of the Initial Posture

When starting a movement the system waits for the user to get into the initial posture. The posture classification method checks the user's current posture until it identifies it as the initial one. These checks are performed in real-time at a rate of about 30 checks per second which is the frequency with which Kinect provides data. When the initial posture is identified the system starts the trajectory recognition.

3.3.2 Trajectory Recognition

The trajectory recognition method has as a main purpose to recognize if the movement itself is well performed. During the recognition, the trajectory of each relevant limb $i$ involved in the movement is compared to the trajectory of the same limb stored for that movement and a similarity value $v_i$ is obtained based on distances between them. If the distance is less than a threshold value $trth_i$ the trajectory path is considered to be correct, and incorrect in opposite case.

Another important aspect here related with the goal of recognizing trajectories in real-time is the frequency of the trajectory recognition or, in other words, how often this comparison among performed and stored trajectories has to be executed. Taking into account that trajectory recognition in real-time is a requirement, it is not pos-
possible to compare the completely performed and stored trajectories only once at the end. For that reason, we also introduced partial trajectory recognition analysis. Therefore, our trajectory recognition method periodically compares for each limb, the trajectory path performed up to that moment by the user with the corresponding stored trajectory. And, as the user may have not finished the movement completely, a last comparison with the complete stored trajectory also has to be executed. In summary, a two-phase analysis takes place: an analysis of partial trajectories and an analysis of the complete trajectory. The trajectory is classified as incorrect when either some partial trajectories or the complete one is incorrect, and as correct in opposite case. In section 5.1.2 we explain how we have obtained the trth trade-off value. Notice that this method is able to detect incorrect trajectories in real-time and can indicate to the user which limb position must be corrected.

In order to calculate the distances among trajectories, the values, we use a variant of the Dynamic Time Warping (DTW) algorithm (please refer to [24] for detailed information on DTW). Although other alternative techniques such as Hidden Markov Models (HMM) have been extensively used for gesture recognition, we chose the DTW technique after analyzing some works that compare their behavior [25–27] and finding that it allows us to: 1) deal with a much smaller training set [25]; 2) not have to re-train a model after a new movement is recorded, an advantage that makes the recording of exercises clearer, simpler and faster for the physiotherapist; and 3) analyze the data in real-time as its performance is high enough [26] for the analysis of exercises.

### 3.3.3 Identification of the Final Posture

While analyzing the trajectories, the exercise recognition method also checks the posture of the user. When the final posture is identified the movement is finished. If an exercise has more movements the method tries to identify the initial posture of the next movement.

Identifying the final posture has a peculiarity given the context of rehabilitation. In some stages of therapy what is expected from the user is to try to reach that position or, at least, to make the physical effort to reach it. Assigning adequate exercises is the physiotherapist’s decision but we also considered a “reach and hold” objective for the patient. Thus, the method adapts the threshold depending on the time spent performing the movement. The initial threshold \( pth_0 \) is multiplied by a flexibility factor \( ff \) that makes the algorithm be less rigid in posture classification. That is to say that the new threshold value is \( pth = pth_0 \cdot ff \).

The flexibility factor \( ff \) is a function that depends on the time spent \( t \) and the time \( r \), in which the movement was recorded, \( ff = 1 + \alpha \frac{t}{tr} \) where \( \alpha \) could be adjusted by the therapists (\( \alpha = 0 \) means no flexibility at all).

### 3.3.3 Exercise Rating

When the user has completed a movement, the method analyzes the result and rates the overall performance. This rate \( r \) is calculated from the values obtained for each relevant limb (as explained in section 3.3.2) with the following formula:

\[
r = \sqrt[2]{v_1^2 + v_2^2 + \ldots + v_n^2}
\]

where \( n \) is the total number of relevant limbs analyzed. Although the flexibility factor \( ff \) does not appear explicitly in the formula, the rate \( r \) takes it into consideration implicitly, because \( v_i \) values will be greater when the final posture is not performed exactly. Finally, the overall exercise rating is the average of the \( r \) rates.

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Training and test sets composition for trajectories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Nº</td>
</tr>
<tr>
<td>ToHeadLeft (THL)</td>
<td>4</td>
</tr>
<tr>
<td>ToHeadRight (THR)</td>
<td>4</td>
</tr>
<tr>
<td>ToHeadForward (THF)</td>
<td>6</td>
</tr>
<tr>
<td>ToRHandUpLeft (TRHUL)</td>
<td>6</td>
</tr>
<tr>
<td>ToRHandDownLeft (TRHDL)</td>
<td>6</td>
</tr>
<tr>
<td>ToRHandUpBack (TRHUB)</td>
<td>6</td>
</tr>
<tr>
<td>Total</td>
<td>32</td>
</tr>
</tbody>
</table>

![Figure 3](image-url) Descriptor classification accuracy depending on threshold
of all the movements that compose the exercise.

4. Setting up the Experiments

We conducted several tests to check the reliability of the algorithm when identifying postures and exercises, as well as to verify that it was capable of processing data in real-time. Moreover, an important issue was to obtain an efficient algorithm with the supervision of physiotherapists, some datasets to validate the performance of the algorithm. Moreover, a physiotherapist recorded the postures and movements for the clinical trials.

4.1 Algorithm Validation Set-up

The datasets created to validate the algorithm contain body postures and recordings of some rehabilitation exercises. In particular, the recorded exercises are part of two therapy protocols. One is oriented to cervical disorders and the other one is oriented to shoulder disorders. These protocols describe with detail the rehabilitation phases and exercises adequate for each treatment. We used six exercises to test our algorithm (The specifications and the execution descriptions of the exercises can be found in this URL). Five healthy volunteers (3 male and 2 female) with ages from 25 to 58 took part in the recording of the above mentioned exercises. Using the resulting data, posture descriptors were annotated manually with each corresponding posture class (seven known posture classes and another one for unknown postures). Those annotated descriptors constituted the test dataset of 4500 different posture descriptors. In addition to this dataset, a training set was created which has 45 posture descriptors labeled with the previous 7 known classes. Table 2 shows the distribution of the posture descriptors on each of the datasets.

To measure the time performance we needed datasets with different sizes. We used six datasets with 45, 4500, 15,000, 20,000, 35,000 and 45,000 posture descriptors respectively in order to perform time measurement tests. The last four datasets are synthetic sets created by repeating the descriptors in the dataset with 4500 descriptors.

We also created two datasets to carry out the trajectory tests. One was used as training set that contained 32 correctly performed trajectories, and the other one was used as test set that contained 48 trajectories, 24 correct and 24 incorrect (Table 3).

4.2 Clinical Trials Set-up

In order to prepare the clinical trials two main tasks took place, the recording of exercises and the selection of real users. With regard to the first task, a physiotherapist recorded a set of exercises to be executed by real users with shoulder disorders. She recorded 8 postures and 6 movements (these 6 movements where reversed making a total of 12 movements) and using our managing tool she combined them into 6 different exercises. The recorded movements plus the reversed version of them were the following: shoulder abduction (1–2), hands to mouth (3–4), shoulder extension (5–6), shoulder flexion (7–8),

Table 4 Posture confusion matrix for threshold 30

<table>
<thead>
<tr>
<th>Posture</th>
<th>Unk</th>
<th>Neu</th>
<th>HL</th>
<th>HR</th>
<th>HF</th>
<th>RHUL</th>
<th>RHD</th>
<th>RHDA</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown (Unk)</td>
<td>802</td>
<td>165</td>
<td>20</td>
<td>18</td>
<td>34</td>
<td>9</td>
<td>29</td>
<td>13</td>
<td>1090</td>
</tr>
<tr>
<td>Neutral (Neu)</td>
<td>29</td>
<td>1223</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1253</td>
</tr>
<tr>
<td>HeadLeft (HL)</td>
<td>0</td>
<td>0</td>
<td>248</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>248</td>
</tr>
<tr>
<td>HeadRight (HR)</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>335</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>337</td>
</tr>
<tr>
<td>HeadForward (HF)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>326</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>326</td>
</tr>
<tr>
<td>RHandUpLeft (RHUL)</td>
<td>33</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>413</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>446</td>
</tr>
<tr>
<td>RHandDownLeft (RHD)</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>341</td>
<td>0</td>
<td>0</td>
<td>346</td>
</tr>
<tr>
<td>RHandUpBack(RHUB)</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>449</td>
<td>0</td>
<td>454</td>
</tr>
</tbody>
</table>

Figure 4 Trajectory classification accuracy depending on threshold

Table 3

<table>
<thead>
<tr>
<th>Posture</th>
<th>Correct</th>
<th>Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neutral</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>HeadLeft</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>HeadRight</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>HeadForward</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>RHandUpLeft</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>RHandDownLeft</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>RHandUpBack</td>
<td>24</td>
<td>24</td>
</tr>
</tbody>
</table>

Five healthy volunteers (3 male and 2 female) with ages from 25 to 58 took part in the recording of the above mentioned exercises. Using the resulting data, posture descriptors were annotated manually with each corresponding posture class (seven known posture classes and another one for unknown postures). Those annotated descriptors constituted the test dataset of 4500...
hands to head (9–10), and shoulder rotation (11–12). These recordings were considered as the ground truth for our algorithm.

With regard to the second task, 15 real users suffering from shoulder disorders were selected to take part in two trials (7 in the first trial and 8 in the second one). They had an average age of 66 in a range from 44 to 83 and they had been doing rehabilitation sessions for at least one month.

5. Experimental Results

In this section we present the experimental results that helped us, on the one hand, to tune the exercise recognition method and, on the other hand, to validate it in clinical trials.

5.1 Tuning the Exercise Recognition Method

In this subsection we explain how the previously mentioned \( pth_0 \) and \( trth \) thresholds were calculated and the feasibility of the real-time processing.

5.1.1 Posture Threshold \( pth_0 \)

As stated in section 3.2, the optimal value for the \( pth_0 \) must be empirically found. A series of tests were conducted with threshold values between 5 and 50 to assess which of them gave the best results. The 4500 posture descriptors of the test set were classified with different threshold values. The results showed that the maximum is reached on threshold \( pth_0 = 30 \) with an accuracy of 91.9% and that with higher threshold values accuracy slowly decreases as shown in Figure 3. As \( pth_0 \) is a trade-off value, then greater or lower values decrease accuracy, but in a different way: with greater values “unknown” posture descriptors are classified as known postures, but with lower values some of the known postures are classified as “unknown”.

The confusion matrix in Table 4 provides more detailed information of these results for the optimal threshold value 30. Each element indicates the number of times the posture of the row has been classified as the posture of the column. The posture descriptors labeled as “unknown” are mostly transitional, undefined postures that occur when moving from one known posture to another.

Notice that most classification errors for unknown postures are produced because they are classified as “neutral” postures. The “neutral” posture is present in all the exercises analyzed, making the transition to it very common.

5.1.2 Trajectory threshold \( trth \)

We calculated the trajectory threshold using a similar procedure to the one used for the posture threshold. A series of tests were conducted with threshold values between 1 and 15. The 48 trajectories of the test set were classified with different threshold values. The results showed that the maximum is reached on threshold \( trth = 10 \) with an accuracy of 93.75%, as shown in Figure 4. With higher threshold values the accuracy decreases because more incorrect trajectories are classified as correct.

Nevertheless, as mentioned in section 3.3.2, a trajectory is classified as correct or incorrect after applying a two phase analysis: a partial trajectory analysis and a complete trajectory analysis. In Table 5, we show the accuracy results obtained after applying the partial trajectory analysis using threshold \( trth = 10 \) (where global accuracy is 89.58%). It’s important to remember that trajectories classified as incorrect during the partial trajectory analysis are definitely classified as “incorrect”.

<table>
<thead>
<tr>
<th>( THL )</th>
<th>Cor</th>
<th>Inc</th>
</tr>
</thead>
<tbody>
<tr>
<td>( THR )</td>
<td>Cor</td>
<td>Inc</td>
</tr>
<tr>
<td>( THF )</td>
<td>Cor</td>
<td>Inc</td>
</tr>
<tr>
<td>( TRHUL )</td>
<td>Cor</td>
<td>Inc</td>
</tr>
<tr>
<td>( TRDDL )</td>
<td>Cor</td>
<td>Inc</td>
</tr>
<tr>
<td>( TRHDA )</td>
<td>Cor</td>
<td>Inc</td>
</tr>
</tbody>
</table>

Table 5: Partial trajectory analysis accuracy

<table>
<thead>
<tr>
<th>Ident. as correct</th>
<th>Ident. as incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct trajectories</td>
<td>91.67%</td>
</tr>
<tr>
<td>Incorrect trajectories</td>
<td>12.50%</td>
</tr>
</tbody>
</table>

Table 6: Complete trajectory analysis accuracy

<table>
<thead>
<tr>
<th>Ident. as correct</th>
<th>Ident. as incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct trajectories</td>
<td>100%</td>
</tr>
<tr>
<td>Incorrect trajectories</td>
<td>33.33%</td>
</tr>
</tbody>
</table>

Table 7: Overall trajectory analysis accuracy

<table>
<thead>
<tr>
<th>Ident. as correct</th>
<th>Ident. as incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct trajectories</td>
<td>91.67%</td>
</tr>
<tr>
<td>Incorrect trajectories</td>
<td>4.17%</td>
</tr>
</tbody>
</table>

Table 8: Trajectory confusion matrix for threshold \( trth = 10 \)

<table>
<thead>
<tr>
<th>( THL )</th>
<th>( THR )</th>
<th>( THF )</th>
<th>( TRHUL )</th>
<th>( TRDDL )</th>
<th>( TRHDA )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cor</td>
<td>Inc</td>
<td>Cor</td>
<td>Inc</td>
<td>Cor</td>
<td>Inc</td>
</tr>
<tr>
<td>Cor</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Inc</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>Tot</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

For this analysis, we have assumed that an incorrect partial trajectory has to be recognized as incorrect for at least 1.5 seconds in order to be definitely classified as incorrect.
The trajectories classified as “correct” by using the partial trajectory analysis do still have to pass the complete trajectory analysis. After that, as can be seen in ▶ Table 6 all the correct trajectories are again (and definitely) classified as correct by the complete trajectory analysis, and 66.67% of the remaining incorrect ones are now well classified.

In ▶ Table 7, we can see the overall trajectory analysis accuracy results corresponding to the combined method of partial and complete trajectory analysis that provides a global accuracy of 93.75%, and in ▶ Table 8 the detailed confusion matrix can be observed.

5.1.3 Testing Real-time processing

Previously, we stated that the proposed algorithm should be able to process Kinect data in real-time in order to give feedback to the user as they were performing the exercise. Kinect provides 30 frames per second so the algorithm had to analyze 30 skeletons in less than a second to avoid execution delays. Posture analysis, which is done continuously, also implies generating the corresponding descriptors to compare with those already stored.

In order to obtain the processing time and establish how many postures can be processed in real-time, we conducted some tests with different dataset sizes.

The tests for time measurement involved loading six datasets with, 45, 4500, 15,000, 20,000, 35,000 and 45,000 posture descriptors respectively.

In ▶ Figure 5 we can observe the average time (in seconds) to process 30 unknown posture descriptors against each of the datasets. The linear regression fits the data well.
data obtained well, so it’s safe to say that the time required to process a posture descriptor increases linearly with the size of the dataset. According to these results the size limit beyond which it would not be feasible to process a dataset in real-time would be around 22,000 posture descriptors, what ensures that it is possible to manage an adequate number of postures in this context.

With respect to the real-time processing of trajectories, as mentioned before, the DTW algorithm is applied. According to [26] it is possible to process more than 10,000 time series in real-time using DTW. In our case, we have just confirmed that it is possible to process the time-series of all the limbs with a frequency of 30 times per second which corresponds to the maximum quantity of data that Kinect can provide. However, through these experiments we also found that using a frequency greater than 3 times per second did not produce significant changes in the results given by the DTW trajectory analysis.

5.2 Exercise Validation with Clinical Trials

In this subsection we show the results obtained from two trials we did in a rehabilitation center managed by Matia Foundation [28]. First of all, we present the dataset used in order to validate the exercise recognition method. After that we present the accuracy of the method when recognizing movements and exercises performed by the users.

The two trials were supervised continuously by physiotherapists that assessed the correct or incorrect execution of the exercises. Therefore, a dataset of annotated exercises was built. While analyzing the execution of the users we found that on average they made 19% of the exercises incorrectly. In Figure 6 the error distribution for each of them is shown. In particular, users 10 and 11 highlight over the others because they get the highest rate of correct execution (96.30% and 100% respectively). These patients had been in rehabilitation for longer than any other. In the opposite side we can highlight patient 7, who could not see the screen well and did not follow the guidance that the 3D avatar provided. In total these patients completed 559 movement executions.

Once the test set was built, the validation of the recognition algorithm was conducted. In the following paragraphs we present the accuracy results grouped by: a) movement; b) exercise; and c) user.

The average recognition accuracy was 95.16%. Out of all of the correctly executed movements, 97.12% were recognized as correct, but the rate decreases to 86.91% when classifying incorrect movement as incorrect. Moreover, in Figure 7 (graph on the left) we can observe that accuracy of Mov4 and Mov10 is 58.32% and 75% respectively. This is because Mov4 and Mov10 are influenced by their initial postures which require lifting the arms towards the head, and in these postures Kinect has difficulties finding joint positions and produces noise in the data. For all other movements the accuracy was above 85%

We want to mention that, for exercise 5 (Figure 7, graph on the right) the accuracy was significantly lower (81.23%), due to the fact that movements Mov4 and Mov10 are part of this exercise.

Finally, while analyzing the accuracy results for each user (Figure 8 we show the accuracy distribution for the users of the second trial) we found that, in general, the average accuracy was consistent with the previous results. However, there was an exception, user 13 (with a 75% accuracy) was wearing a loose blouse that made it difficult for Kinect to recognize joints correctly.

6. Discussion

Several works have documented the effectiveness of using different kinds of telerehabilitation systems at home [4–6]. The current trend is oriented towards the development of systems that make use of non-invasive devices and, in particular, the core device of many proposals is Kinect for Windows [17–21], because it is a low-cost portable tracking alternative which does not require users to wear specialized equipment for tracking. A limitation that Kinect presents is that skeleton recognition works well when the user is facing the device, but lateral recognition is not accurate.

Developing a complete functional telerehabilitation system based on Kinect has revealed some important considerations. For the system to be easily adopted by
users, it must provide an interface that users find fun and at the same time allows them to notice the errors they make and their progress throughout the treatment. In this sense, the interface provided by our system includes motivational features such as two 3D avatars that show the user how to execute the exercises and the actual execution respectively (so she/he can be aware of the differences). Moreover, the interface also includes some elements that provide information about the ongoing therapy session. With this interface we realized that the system empowers and keeps the user aware of his/her therapy, but also provides a game-like immersive experience that motivates and makes the therapy more enjoyable.

From the point of view of therapists, after having worked with them closely, we realized that they appreciate the fact that the system follows the guidelines of their usual way of working and so our interface presents a menu as a datebook that shows information in it. Moreover, a relevant issue is the way new exercises are added to the system. The proposed system can be loaded with exercises for a wide variety of physical alterations. The system allows therapists to perform and record the new exercises themselves. In addition to recognizing exercises, the data gathered by our proposal can be used for other purposes in the context of telerehabilitation.

From the data provided by Kinect we have shown that it is possible to develop an efficient and reliable exercise recognition algorithm.

In the clinical trials, we obtained 95.16% accuracy in exercise recognition. There is no reference benchmark to make an accurate comparison with other recognition algorithms. However, our accuracy results are comparable to those obtained by other works that provide solutions in the rehabilitation area. Among them, we can mention [12, 22, 29]. In [12] Spina et al. reported a 96.7% accuracy but using a smartphone’s build-in inertial sensors to monitor exercise execution. In [22], Gabel et al. present a gait analysis system based on Kinect sensor that provides correlation coefficients between the Kinect-based prediction and the true value greater than 0.91 for both arms. In [29], the authors present a system for cognitive rehabilitation that achieves a successful monitoring percentage of 96.28%.

Regarding the clinical trials we consider that the collaboration with the Matia Foundation gave us a relevant insight of our proposal, not only for the results obtained in exercise recognition, but also for the experience with the physiotherapists and the patients that took contact with our system. In these medical trails patients did a 19% of the exercises wrong. It seems to be a high rate of failure, but we want to highlight that for all of them it was the first time interacting with Kinect and also that our patients were elderly people not used to interacting with computers.

### 7. Conclusion

In this paper we have presented a Kinect-based algorithm for the monitoring of...
physical rehabilitation exercises. That algorithm recognizes the main components of the exercises, postures and movements in order to assess their quality of execution. Furthermore, the friendly interface that it supports provides end users with a gamelike immersive framework. This framework motivates them and makes the rehabilitation sessions more enjoyable while at the same time it allows users to be aware of their progress. Moreover, using that interface the physiotherapists can define in an easy way a great variety of exercises customized for users. With respect to technical issues, the algorithm is capable of making real-time recognitions of the exercises and, furthermore, its behavior is good using only a few samples in the training step. Finally, the feasibility of the algorithm has been validated in a real scenario with 15 users achieving a monitoring accuracy of 95.16%. This performance was considered very adequate by the physiotherapists that supervised the clinical trials. In future research we expect to analyze our algorithm using the upcoming version of Kinect and to develop a data mining module to exploit the raw data gathered by the system in order to extract meaningful and actionable information for the physiotherapists and users.

Acknowledgments

The work of D. Antón was supported by the Spanish Ministry of Economy and Competitiveness under Project TIN2013-46238-C4-1-R and by the Basque Country Government financial grant IT797-13. We thank the staff of the Matta Foundation and their patients for collaborating in the trials presented in this paper.

References


© Schattauer 2015

Methods Inf Med 2/2015