International Journal of Computer and Information Technology (ISSN: 2279 – 0764)
Volume 03 – Issue 04, July 2014

Detection of Pilates Exercises Based on Movement Sensors Data in Modern Smartphones

For children aged 11 to 13

Manuel Parfant
Institute for Information Systems and Computer Media
Graz University of Technology
Graz, Austria
Email: m.parfant {at} student.tugraz.at

Martin Ebner
Institute for Information Systems and Computer Media
Graz University of Technology

Dietlind Deutschmann
Medical University Graz
Graz, Austria

Abstract— **Nowadays more and more children are suffering from back pain and postural defects. Different studies dedicate to this issue and various workouts are specially designed for prevention. Generally a therapeutic success remains low, because the patients do not accomplish the required exercises at home or perform them wrong and the therapist has no possibility to monitor their activities. This publication aims to answer the research question, if it is possible to monitor the activities of children, just with the data provided by sensors included in their personal smartphones.**

Keywords-motion detection; pattern recognition; monitoring; sensors; smartphones

I. INTRODUCTION

Nowadays back pain due to postural defects is one of the most frequent chronic symptoms in our population. Particularly the number of school children with postural defects is increasing [12]. The earlier a weakness in posture is recognized, for example influenced by muscular imbalance, the higher the chances to prevent further defects [13]. But the arrangements to preventions are still insufficient. Latalski and his colleagues request that „There is a need for the creation of a system of education for parents and children concerning postural defects and risks resulting from these defects“ [12].

For this reason a Pilates-program, called Wiraculix workout, was constructed for regular sports exercises to prevent children or adults of back pain. Some of the exercises will be described as following:

“Criss-Cross”: Lay back on your back and hold your feet up bending your knees at an angle of 90 degrees. Put your fingers to your temples and press your elbows apart. The elbows should stay in this position during the whole exercise. Take your head off the floor and start with the exercise: Put one elbow to the knee on the other side while you are bending this knee and stretching the other leg. Then you are going on the same way while switching the sides.

“Crawler”: Lay back on your back and but your feet on the floor. Lift up your bottom until your stomach built a line with your thighs. Try to hold your body in this position a few seconds, then lay your bottom slowly down to the floor and continue with the same exercise.

“Rolling like a ball”: Sit down on the floor and bend your knees. Put your chin to your breast and try to be as small and as round as possible in your back. Now hold in your stomach, roll back and forward and don’t put your feet on the floor anymore.

“Air mattress”: Lay down on your stomach and stretch your legs as far as possible. Put your arms on the floor and lay down your head. Lift your legs a little bit off the floor and move them up and down frequently.

“Superman”: Put your knees and hands on the floor, hold in your stomach and try to put your back in a straightly position. Lift one arm and the leg on the other side and stretch them as far as possible. Try to hold this position for a few seconds, then move your arm and leg down to the floor and lift the other ones.

“Spine Twist”: Sit down on the floor with legs stretched or cross-legged, but put your back in a straight position. Put your arms to the sides and stretch them. Now begin to turn your whole trunk to one side while holding the arms in the start position without changing the angle. Turn slowly to one side as far as possible, than continue to the other side.

Often therapeutic success remains low because especially children do not accomplish the prescribed exercises from the therapist at home or perform them wrongly. [1]

The research question of this publication is if it is possible to monitor the exercises which are done at home by the children, in a straightforward way. The main target groups of this research study are children and teenagers and therefore it is very important not to use heavy, awkward and expensive wearable computing systems. Due to this fact today’s smartphones and their included sensors are used to recognize these exercises and replace wrong and incomplete therapy protocols. Another fact is that the dissemination of smartphones is increasing rapidly and get widely spread and the pupils are familiar with such devices [1].

II. METHOD

To detect motion exercises from children at home a prototype, which consists out of an iOS-Client, a webservice and a webapplication is implemented, using the classic waterfall software engineering process [14].

The main requirements for the system are:

- *Usability*
  Because the system is mainly intended for children aged 11 to 13, it has to be very easy to use and therefore it has to have an almost intuitive usability.
- *Fast response*
  Because children are usually impatient, the response of the system has to be fast; nobody wants to wait for minutes to receive the evaluation.
- *Easy to understand evaluation*
  The resulting evaluation should be easy to understand. A graphically solution is required.
- *Collect data*
  The gained data should be stored for following studies and analyzes.
- *Calculate interesting indicators*
  The backend should calculate interesting indicators for the therapist and provide a graphically clear presentation of them.

For this purpose, a three-tier architecture with the iOS-Client and a webservice for the classification task respectively a web application is designed. To ensure the best possible accessibility, Google's App Engine platform and data store [8] is chosen. To provide fast feedback, the client should transfer the data parts also during the workout to the web service, which could evaluate it asynchronously.

Fig. 1 presents an overview of the implemented prototype, called MotionTracker-system. The different components and their interactions are shown and explained in the next paragraphs.

Before the children start their workouts, they have to attach the device to a reference point on the body. This can be done with a bag which is designated for sport activities. Fig. 2 shows a kid wearing the device on the reference point for the exercise "Circus-Horse".

The iOS-Client collects the motion data, delivered from the integrated sensors of the iPhone, with a specific sampling rate (50 hertz). Furthermore, it sends this data to the MotionTracker-webservice which returns the evaluation of the workout.

The corresponding web service provides the API commands START - COLLECT - STOP and STORE, which are explained in section III.B. Uploaded data is analyzed by the

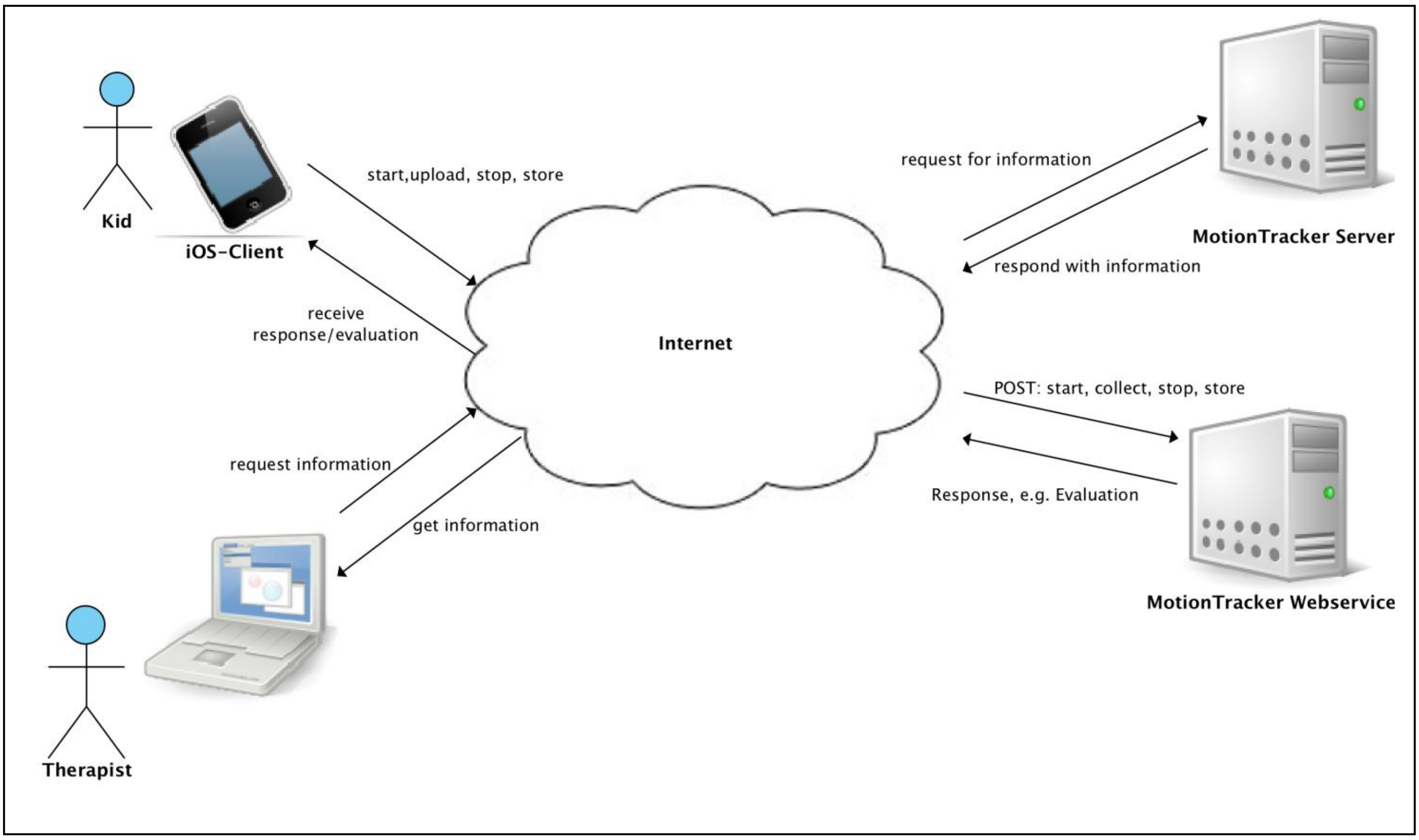

Figure 1: An overview of the MotionTracker-system

webservice which provides an evaluation counts the repetitions and stores everything for further analyses. After finishing the

Figure 3: Kid wearing the device during a workout

sports program and the children get the evaluation of the recorded movements.

Furthermore the webapplication gives the therapist the possibility to monitor the activities of the subjects in an easy and fast way. Recorded data is represented graphically and also in a detailed way; underflows of specific limits are particularly marked. Furthermore, the therapist gets additional information about the rate of participation and precision of the movements.

This implemented system has been evaluated using four test children with different prerequisites. The results of the evaluation are discussed in chapter evaluation.

III. TECHNICAL IMPLEMENTATION

A. *The MotionTracker-app (iOS-Client)*

The iOS-client, called MotionTracker-app, has to be started on the children's device before the workout. This device has to be worn during the workout, using special designed bags. When the child starts with the Pilates exercises, it has to signalize it to the app with a simple touch on a start button. Using the START command of the webservice the MotionTracker-app notifies it about the start (see section B). After that the client begins to collect the gyroscope's and accelerometer's data with a sampling frequency of 50 hertz.

From a technical perspective, this is done with the help of Apple's *CMMotionManager*-class. To ensure that no sampling point is lost, a push-approach is used. This means that after every sampling a code block gets executed in a separate thread, no matter if the device is busy at the moment or not. [2]

The collected data is sent to the MotionTracker-server during the workout in the interval of one minute and afterwards. For this purpose, the API commands COLLECT, STOP and STORE are used, which are explained in section B. To stop the workout, a simple touch on a stop button has to be made. The data upload during the workout enables faster feedback from the system for the user. The results delivered from the server are presented graphically and suggestions for improvement are shown. For communication between the components, the data is transferred using JSON-format [7].

Fig. 3 shows the workout view of the MotionTracker-app. The start button is shown, as well as all received evaluations. Any evaluation consists of the different exercises and their found repetitions. Another feature is the calculated accuracy, which is indicated by the green percentage filled checkmark. If one touches an exercise, a detailed evaluation is shown, where the individual repetitions, their accuracy and potential suggestions for improvement are displayed.

B. *The MotionTracker-Webservice*

This webservice is a Java Servlet running on the Google App Engine [8]. It provides four different API Commands: START, COLLECT, STOP and STORE.

- *START*
  This command creates a session for a user. This is necessary to map received data to specific users in following commands.
- *COLLECT*
  The collect command takes intermediate data parts of the actual workout, stores it temporarily and initiates the classification of the data. The classification process is explained in section F. The mapping and intermediate analysis is necessary to provide fast feedback to the user because after stopping, the system has to analyze the last interval just in time.
- *STOP*
  The stop command takes the data of the last interval, stores it temporarily, initiates the classification and sends summarized feedback to the client afterwards.

Figure 2: Workout view of the MotionTracker-app

- *STORE*
  The STORE command stores the actual session permanently to the database.

C. Algorithm overview

To recognize the different exercises in the gained data stream the following algorithm is used:

- *Preprocess gained data using signal averaging*
- *Extract samples using peak detection on simple patterns*
- *Extract features out of resulting data matrix*
- *Classify samples using earlier trained support vector machines*
- *Post process found samples, i.e. delete duplicated entries*

The different steps are explained in the following chapters.

D. Used sensors and information

Since the 4th generation, the Apple iPhones are shipped with two sensors, the accelerometer, which measures the velocity (m/sec), and the gyroscope, which measures the rotation angles of the device (rad/sec). Both supply data for the three-dimensional space with a specific sampling rate. [2]

For each of the exercise a fixed reference point is defined which is fundamental for its movement and to classify the gained data. The reference point should capture the most important movements of an exercise and changes from exercise to exercise.

Fig. 4 shows the rotation rate over the time, which is delivered from the gyroscope sensor, of the exercise “Spine-Twist”. The red curve corresponds to the rotation rate around the z-axis, which has a significant amplitude in this movement. On the other hand, the rotation rates around the green x-axis and the blue y-axis are relatively low.

Four different executions of the “Spine-Twist” exercise are shown in Fig. 5. The first is well done; it can be pointed out

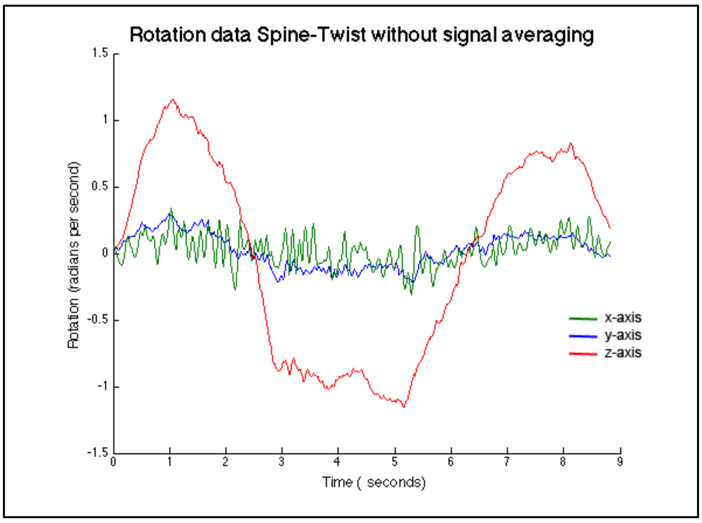

Figure 4: Rotation data over the time from the exercise "Spine-Twist"

that the z-dimension has a significant movement while the other dimensions’ movements are relatively low. The second trial is done with turned arms. Executions three and four are typical wrong ones. The child performed these two accomplished repetitions without keeping the arms straight. It can be pointed out that this results in higher movement also in the x- and y-dimensions, which is the result of the different angle of the arms.

E. Preprocessing and segmentation

Before the data is sent to the classifier, a preprocessing step is done. To reduce “noise” in the data, which is for example because of inexact movements, and to remove unnecessary peaks a signal averaging is performed. This reduction is shown in (1). The index i indicates the actual point, whose value is calculated. Variable n is the amount of neighbored sampling points which influence the value for point i in the calculation, e.g. two means, that the two points before and two points after this point i are considered. N indicates the total number of available sampling points. In summary, this equation calculates

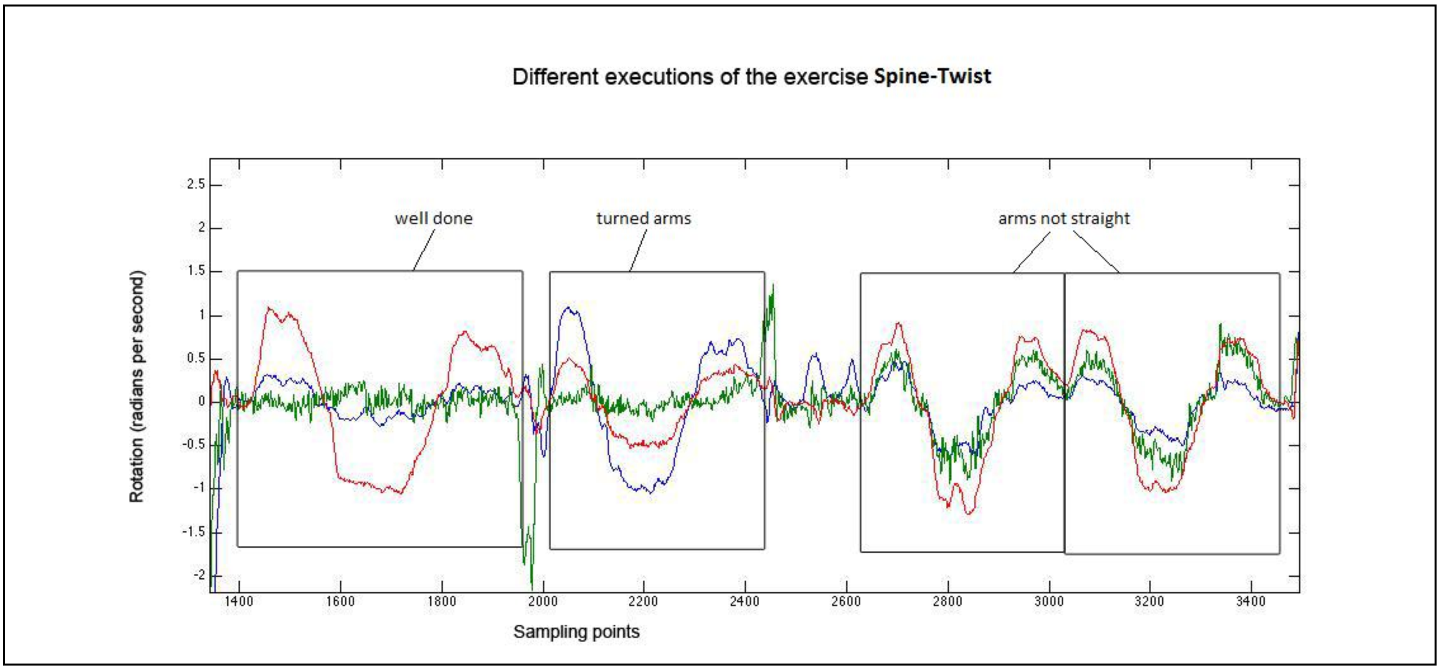

Figure 5: Different executions of the exercise "Spine-Twist"

the mean of the points within the range [*i-n* to *i+n]*, which will be the new value for point *i*.

$$x(i) = \frac{1}{2n+1} \sum_{j=1-n}^{i+n} x(j)\,,\; n < i < N - n \qquad (1)$$

The segmentation of the data stream to individual repetitions of the exercises is done by a simple peak detection. The data stream is searched through for distinctive patterns using the signal averaged data. Each exercise has a significant movement in one axis which is used for matching. To distinguish between real exercise repetitions and randomly matched movements by the peak detection the classifier is used. In a second classification step, the repetitions are assigned to specific error classes, which are explained in section F.

Using these gained data sets, a data matrix which consists out of the six columns can be built. These columns are the three dimensional rotation data and the three dimensional acceleration data. The count of rows depends on the time interval. The construction of this data matrix is shown in Fig. 6.

F. Classifying methods

The gained data matrix can be used for classification. According to Duda, Hart and Stork [3], a typical classifier consists of five components - collection, segmentation, feature extraction, classification and post processing. The segmentation can be done with a simple peak detection on the gyroscope data. Each exercise has a significant shaping which can be used.

From these resulting samples different features for the classification step are extracted. These features should be as distinguishable as possible to improve the classification. [3]

Li Kulkarni and Prabhakaran suggest in [4] to use singular values [10] for classifying motion patterns, which are used in this work. Furthermore classical features like the mean (expected value) and the standard deviation from each of the different dimensions, as well as cross-correlations, between them and between the whole three-dimensional acceleration and rotation data, are used.

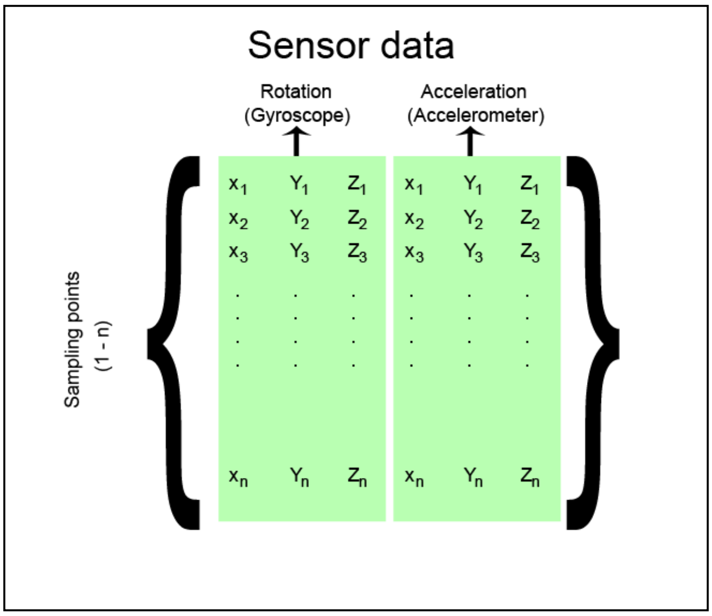

Figure 6: Construction of the data matrix used for classification

Equation (2) shows the calculation of the expected value $\bar{X}$ of the sample, where *N* is the total amount of considered points, *i* is an index running from *1* to *N* and x_i is the actual point at index *i*.

$$\bar{X} = \frac{1}{N} \sum_{i=1}^{N} x_i \qquad (2)$$

The calculation of the standard deviation σ^2 is shown in (3). $\bar{X}$ is the expected value as explained above. Again, *N* is the total amount of considered points, *i* is an index running from *1* to *N* and x_i is the actual point at index *i*.

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{X})^2 \qquad (3)$$

The cross-correlation is calculated as normalized cross-correlation E_{NCC} [5], which is shown in (4). *u* is the starting index of the considered data segment. I_0 and I_1 are the two compared data segments, and $\bar{I_0}$ and $\bar{I_1}$ their expected values. Index *i* indicates the actual position in the segment of the calculation.

$$E_{NCC}(u) = \frac{\sum_i [I_0(x_i) - \bar{I_0}][I_1(x_i+u) - \bar{I_1}]}{\sqrt{\sum_i [I_0(x_i) - \bar{I_0}]^2} \sqrt{\sum_i [I_1(x_i+u) - \bar{I_1}]^2}} \qquad (4)$$

The classification of the gained features is done with support vector machines. The different categories are interpreted as exercises in this context. In a first step, a support vector machine assigns the sample to the specific exercise; in a second step several support vector machines assign the movement to specific error classes. For this purpose a set of 25 features as mentioned above, and a linear kernel function is used.

The support vector machines are implemented as “one vs. all” classifier, which means that the training samples of the specific classes are positive and that from the others are negative. [9]

The features are also scaled to receive a higher accuracy and more stable results, which is done as shown in (5). [6]

In (5) x_{norm} is the scaled value and $x_{original}$ is the original value of this feature, $\bar{X}$ is the expected value and σ the standard deviation of all features of this dimension.

$$x_{norm} = \frac{x_{original} - \bar{X}}{\sigma} \qquad (5)$$

Because assigning a repetition to exactly one error class is often practically impossible, overlaps between them are allowed. Fig. 7 shows the error classes of the exercise “Criss-Cross”. The different error classes and also overlaps between them are shown, which means that the child has done several errors.

The accuracy of the accomplished exercises as shown in section A is depending on the assigned classes. A “well done” classified sample has initially 100 percent. If additionally some error classes are classified, the score decreases. If it is just classified as valid, but no specific class is found it decreases to 50 percent.

G. Post processing

Different reasons make a post processing step necessary:

- The MotionTracker iOS-Client is sending data to the webservice during the workout every minute, where these parts get analyzed.
- For some exercises it is possible to start in two directions e.g. turn to left or right first, which produces overlapping examples.
- When the child holds a position for a while, peaks can emerge and therefore produce duplicate repetitions.

To avoid such duplicate entries all found repetitions are ran through again and cleaned. The different repetitions get checked if they are simultaneously relating to the time domain. If the overlapping percentage is greater than a specific threshold, these repetitions are deleted. Equation (6) shows the calculation of the overlapping percentage *p*, where l_A is the length of the overlap between the two repetitions and l_B is the length of the second repetition in terms of time.

$$p = \frac{l_B}{l_A} \qquad (6)$$

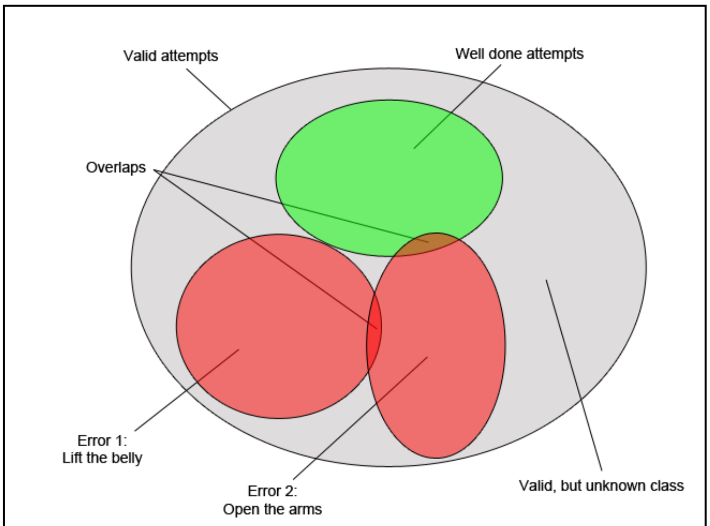

Figure 7: Different error classes of the exercise "Criss-Cross"

IV. Evaluation

The developed prototype was introduced to pupils of two different schools nearby Graz in their physical education lessons. The kids were very interested in the used technology. More detailed evaluations about the introduction in schools will be done in the future. This publication primarily evaluates that our approach is basically working.

Finally it was tested and evaluated with four children, all with different prerequisites concerning the age and the physical level. With the help of especially designed bags, each of these the children got a device, which had the MotionTracker-app running on it, attached. They were told to accomplish ten repetitions of all six exercises of the Wiraculix workout. Practically this was not always possible because the children performed test repetitions, counted wrong or were not able to finish it because the fitness level was too low.

TABLE I. shows the exact repetitions done by the children, which are used for the evaluation.

Additionally the children were filmed during the workout to evaluate the accomplished exercises manually. Thus the number of accomplished exercises or found errors in the evaluation refers to the rating of the therapist.

TABLE II. shows the evaluation of the accomplished workouts. The different classification rates for each exercise and child as well as overall rates of the system are shown.

The hit rate or true positive rate measures the well done repetitions of an exercise, which are also recognized as such. The wrong repetitions are not influencing the hit rate.

In contrast the fallout or false positive rate measures wrong repetitions, which are recognized as well done. Found well done exercises are not influencing the fallout rate.

Collectively the correct classification rate indicates how many repetitions were assigned to the right category. Thus this rate summarizes the hit rate and the fallout rate because it measures the well done and wrong repetitions, which are also recognized as such. All this values are corresponding to the total amount of possible samples found by the segmentation step.

Altogether there were 633 possible repetitions of the different exercises found, 223 were real repetitions and 410 randomly found segments. The implemented system could recognize 189 of the 223 real repetitions, which results in a hit rate of about 84.75 percent. 73 of the 410 randomly found segments were falsely assigned to real repetitions; this corresponds to a fallout rate of about 17.8 percent. Totally the correct classification rate is about 83.1 percent.

The result shows that nearly all real repetitions could be found. Relations between the complexity of the exercises, the amount of movement in it and the hit rate are noticed. More complex exercises, like the exercise “Criss-Cross”, as well as ones with less movement, like the exercise “Caterpillar”, achieved the lowest hit rates. For all others a practically sufficient hit rate of more than 95 percent could be achieved.

TABLE I. ACCOMPLISHED REPETIONS OF THE SUBJECTS

Exercise	Repetitions of the exercise				
	Child				*Overall*
	A	*B*	*C*	*D*	
Criss-Cross	10	12	5	8	35
Spine-Twist	12	10	4	7	33
Circus-Horse	11	11	5	9	36
Rolling-Ball	9	10	2	10	31
Air-Mattress	14	14	14	11	53
Caterpillar	10	10	5	10	35

TABLE II. CLASSIFICATION RATES OF THE EVALUATED WORKOUTS

Exercise	Classification rates				
	Subject				*Overall*
	A	*B*	*C*	*D*	
Criss-Cross					
Hit rate (true positive rate)	100	91.67	80	100	94.29
Fallout rate (false positive rate)	30.43	45.83	30.77	21.88	31.52
Correct classification rate	78.79	63.64	72.22	83.33	75.59
Spine-Twist					
Hit rate (true positive rate)	100	100	100	85.71	96.97
Fallout rate (false positive rate)	10.53	0	0	0	3.77
Correct classification rate	93.55	100	100	94.74	96.51
Circus-Horse					
Hit rate (true positive rate)	100	100	100	100	100
Fallout rate (false positive rate)	0	41.18	33.33	11.11	22.5
Correct classification rate	100	75	87.5	94.44	88.16
Rolling-Ball					
Hit rate (true positive rate)	100	90	100	100	96.77
Fallout rate (false positive rate)	30	40	40	0	31.03
Correct classification rate	84.21	75	71.42	100	83.33
Air-Mattress					
Hit rate (true positive rate)	100	78.57	64.29	90.91	98.11
Fallout rate (false positive rate)	7.41	19.23	16.67	36.84	12.22
Correct classification rate	95.12	80	75	73.33	91.61
Caterpillar					
Hit rate (true positive rate)	10	0	100	80	40
Fallout rate (false positive rate)	0	0	5.56	16.22	6.6
Correct classification rate	71.88	74.36	95.65	82.98	80.14
Summary					
Hit rate (true positive rate)	86.36	77.61	82.86	92.73	84.75
Fallout rate (false positive rate)	12.5	23.08	16.18	18.58	17.8
Correct classification rate	87.07	77.17	83.5	85.12	83.1

Nevertheless it must be taken into account that randomly found samples are assigned to real repetitions too often. More complex exercises and such with less movement have higher fallout rates than others. Furthermore differences between the age and fitness level of the children and the classification rates could be observed. Older children or such with a good fitness level could perform the exercise with a higher accuracy than the younger or less fit ones.

The exact assignment of an accomplished exercise to an error class could be done in 74.32 percent of the samples. A right assignment to an error class means an agreement to the manually made classification using the filmed workouts. The summary of this evaluation is shown in TABLE III.

TABLE III. CLASSIFICATION RATES OF RECOGNIZED ERRORS

Exercise	Recognized errors				
	Child				*Overall*
	A	*B*	*C*	*D*	
Criss-Cross	76.67	69.7	33.33	87.5	66.80
Spine-Twist	70.83	60	75	100	76.46
Circus-Horse	45.45	72.73	80	77.78	68.99
Air-Mattress	67.86	18.18	55.56	80	55.40
Caterpillar	100	0	100	68.75	89.58
Overall	72.16	73.54	68.75	82.81	74.32

V. DISCUSSION

Basically, the prototype allows children to perform exercises at home in a somehow controlled situation.

Although this approach provides good results it is limited, due to the fact that it can only monitor one fundamental part of the movement. Because of that, not all of the movements can be monitored, i.e. legs cannot be recognized when the reference point is on one of the arms.

Therefore this reference point must be defined on a position which is covering the most important parts of the movement. E.g. if the movement consists of rotating the arms, the reference point has to be on one of these.

Furthermore there can be correct results out of incorrect movements, when moving the smartphone around with the hand and therefore constructing similar movement data.

In contrast the problem, that incorrect movements are recognized as real repetitions can happen. Reasons are as already mentioned the complexity of the exercise, the fitness level and age of the subjects or different coordination skills.

The more precise the child can perform the exercises, the better the result.

Also exercises with less movement cannot be controlled with this approach. E.g. if the exercise consists out of holding a position for a while, too less motion data is produced from the sensors to distinguish the exercises appropriate.

With the use of hierarchical steps for classification it is also possible to distinguish between very precise and imprecise repetitions and even typical errors. This enables the possibility to give feedback to the children concerning the accuracy and provide them suggestions for improvement.

Practically the implemented system is very useful. The correct classification rate of the exercises with about 83 percent is good for monitoring issues. Furthermore the correct classification rate of typical errors of the exercises, which is about 75 percent, is useful to provide the kids suggestions for improvement.

VI. Conclusion

The developed system fulfills the requirements, which should monitor children's' activities at home and replacing wrong and incomplete therapy protocols.

The iPhone's integrated sensors, i.e. the gyroscope and the accelerometer, provide a good base for further analyses.

Although it is not yet possible to measure the movements exactly, this is a big step to simplify evaluation and the control of sports programs. Furthermore more kids could participate in the program if the app was extended to android systems.

VII. References

[1] M. Ebner and M. Grimus, "Learning with Mobile Devices Perceptions of Students and Teachers in Lower Secondary Schools in Austria", ED-Media, 2014, accepted, in press.

[2] Apple Inc., "Event Handling Guide for iOS", 2014, https://developer.apple.com/library/ios/documentation/EventHandling/Conceptual/EventHandlingiPhoneOS/EventHandlingiPhoneOS.pdf, 15.02.2014.

[3] Richard O. Duda, Peter E. Hart and David G. Stork, "Pattern Classification (2nd Edition)", Wiley-Interscience, 2000

[4] Chuanjun Li, Punit R. Kulkarni and B. Prabhakaran, "Segmentation and recognition of motion capture data stream by classification", Multimedia Tools and Applications, 2007.

[5] Richard Szeliski, "ComputerVision: Algorithms and Applications", Springer, 2010, http://szeliski.org/Book/, 19.02.2014

[6] Chih-Wei Hsu, Chih-Chung Chang and Chih-Jen Lin, "A Practical Guide to Support Vector Classification", 2010, http://www.scie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf, 19.02.2014.

[7] ECMA International, "The JSON Data Interchange Standard", 2013, http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf, 08.04.2013.

[8] Google Inc., "Google App Engine: Platform as a Service", 2014, https://developers.google.com/appengine, 03.03.2014.

[9] C.M. Bishop, "Pattern Recognition and Machine Learning. Information Science and Statistics", Springer, 2006, http://books.google.at/books?id=kTNoQgAACAAJ.

[10] Kenneth Lange, "Singular Value Decomposition. Statistics and Computing", Springer, 2010, http://dx.doi.org/10.1007/978-1-4419-5945-4_9.

[11] Martin Ebner and Sandra Schön, editors, "Lehrbuch für Lernen und Lehren mit Technologien.", 2011, http://l3t.tugraz.at/index.php/LehrbuchEbner10/issue/current.

[12] Latalski M, Bylina J, Fatyga M, Repko M, Filipovic M, Jarosz MJ, u. a., "Risk factors of postural defects in children at school age", Ann Agric Environ Med. 2013;20(3):583–7.

[13] Newcomer K, Sinaki M, Wollan PC. "Physical activity and four-year development of back strength in children". Am J Phys Med Rehabil. 1997;76:52–8.

[14] Benington, Herbert D., "Production of Large Computer Programs", 1983, IEEE Annals of the History of Computing