Assessing fine motor skills of children through sensor data and game state information

Jörg Sander
10881530
jorg.sander@toologic.com

1 Introduction

1.1 Background

Children's activity levels have decreased over the past 25 years [1]. The amount of physical activity in the pre-school and toddler period is associated with the development of motor skills [3], [4]. Less physical active children engage themselves less in new activities that challenge their motor development. Consequently, the motor development is delayed. In Amsterdam alone the percentage of children with a delay in gross motor development is about 15% and this number increases [2].

In order to counter these developments it would be beneficial if the early detection of delayed motor skill development in primary school children could be facilitated with so called *smart play sets* that can be used in physical education. These play sets are capable to provide a high level diagnosis through analyzed sensor data (e.g. accelerometer or gyroscope) and possibly engage children in physical activities that are fun to do and offer relevant exercises for care and cure. In addition close monitoring of fine motor development could benefit both teachers to correctly identify the next learning step, as the child, in being engaged in a more optimal learning situation.

This article describes the experiments that were conducted in order to assess the motor skills of 6 to 7 years old children (hereafter referred to as children or toddlers) by means of an already developed toy called *futuro cube*[1]. The underlying assumption is that the sensor data and the game state information that the futuro cube provides can be used to evaluate the fine motor skills of children that are necessary in many tasks in the primary school setting for example write, draw, puzzle or craft.

As a first step, the goal was to investigate the feasibility of this assumption and more specific the following research questions were addressed:

1. How accurately can expert rating of fine motor skills of children be estimated from the sensor data and game state information of the futuro cube?
2. What features might prove most sensitive to determine changes in motor capacity of children?

This approach serves the goal of developing features and algorithms that can provide a basis for the development of other (more interactive and playful) game devices that are able to assess the fine motor capacities of children.

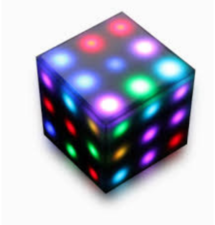
1.2 Related work

This section presents related work on automatic assessment of motor functions from sensor data collected during predefined motor skill assessment tasks. Previous research on quantitative assessment of motor abilities was mostly done on the topic of wearable technology in physical medicine and

[1] http://www.futurocube.com/

project AI (2016), SmartPlaySets, University of Amsterdam.

Figure 1: futuro cube



rehabilitation where subjects were suffering from severe motor impairments e.g. stroke survivors [12] or cerebral palsy [4].

In most related work, the goal was to assess motor skills with either signal processing or machine learning. The ground truth was obtained by experts' ratings or standard clinical rating scales. Using machine learning, the most common approach is to extract features from sensor data and apply a feature selection algorithm in order to reduce the dimensionality of the data. Finally one or more classification algorithms [11] are trained, validated and tested on the available data in order to assess their generalization performance.

With such a machine learning approach motor recovery of stroke survivors has been assessed using accelerometers attached to different positions of the patient's affected arm [12, 13]. Similarly, authors [14, 15] extracted statistical features from accelerometer data to predict motor function scores during different tasks. They were able to predict the scores with an error of 10%.

In [16] a post-stroke patient performed the Wolf Motor Function Test (WMFT) while being equipped with a single inertial measurement unit (IMU) on her wrist. The goal was to distinguish the movements between the affected and unaffected arm of the patient by means of a Naive Bayes classifier that used calculated features from the sensor data.

Zhang et al. [17] provided a fine-grained assessment of motor skills by capturing detailed patterns contained in the patients' movements. They used Dynamic Time Warping to compare movement trajectories of the affected and unaffected arm which embarks on a different motor skill assessment technique than building mapping functions that correlate sensed movement signals to expert or clinical rating scales.

The work presented in this article differs from previous research on quantitative assessment of motor abilities using sensor data in that the children participating in the experiments here do not suffer from severe motor skill impairments due to (birth) injuries. In addition the fine motor abilities of these toddlers were not pre-assessed with standard clinical rating scales. It is assumed that the differences in fine motor skills between these children are more fine-grained compared to the severe impaired children or adults participating in previous research.

Another difference with preceding experiments lays in the fact that the analysis presented here will include so called *game state* features e.g. an error measure that will quantify how accurately a child can perform the predefined task.

2 Method

2.1 Experimental design

A total of 9 children, age 6 to 7 years, from a primary school in Amsterdam participated in the experiments. All children were assessed before the study by certified physical education teachers and 5 of them were classified as having developmental delayed fine motor skills (e.g. difficulties during handwriting). All characteristics of the participants can be found in table 1.

Each child performed six predefined motor tasks during the experiments. Children were tested individually for approximately 30 minutes and started with the futuro cube[2]. The children were carefully instructed before the game and performed the roadrunner game only once (without a

[2] other game devices used during the experiments were: building blocks, balance boards, agility ladder

Table 1: Details on participating children

ID	Age [years]	Gender	Handedness	Assessment
1	7	Boy	Right	delayed fine motor skill development
2	7	Boy	Left	delayed fine motor skill development
3	7	Boy	Left	delayed fine motor skill development
4	7	Girl	Right	
5	6	Boy	Right	delayed fine motor skill development
6	7	Girl	Right	
7	7	Girl	Right	
8	7	Girl	Right	delayed fine motor skill development
9	7	Girl	Left	

training phase). Capturing of the data started synchronously with the actual game. At the end of the experiment the child had to evaluate the different games by means of a smiley scale rating specifically developed for children[3].

2.2 Game device

The futuro cube uses vibration, audio and colorful LEDs to interact with the player. A foto of the cube is shown in figure 1. Inside the cube a 3-axial accelerometer tracks rotation with a sampling frequency of 1.25 kHz. It uses a low energy communication protocol to wireless exchange information with a computer or another cube. Each site of the cube is divided into nine squares, containing a LED each that can be identified by a specific *index*, *side* and *square* number (a map of the cube is shown in figure 3).

2.3 Specifications of accelerometer meter and sensor data

The experimental data is collected from a 3-axial accelerometer (LIS3DH [4]) that is built into the *futuro cube* game device.

The built-in accelerometer uses a scale of $\pm 8g$ with a resolution of ± 2048 (where $1g \approx 256$) and is capable of measuring accelerations with output data rates from 1 Hz to 1.25 kHz. For each of the 3-axis the data output is 16 bit.

The following pre-processing is applied to the raw sensor signal in the cube before it is available in the *game software*:

- the signal is *right shifted* by four bits in order to attenuate high noise;
- a moving average filter of size four is applied to the signal;
- the filtered signal is passed to the *game software* every 8 ms.

Therefore the accelerometer data available in the game software has a maximum sampling frequency of 125 Hz.

2.4 *Roadrunner* game

The futuro cube can be programmed by a simple, typeless 32-bit language called *PAWN*[5]. The company that developed the futuro cube delivers a couple of APIs written in PAWN that facilitate the creation of new games. For this work a game application was scripted (referred to as roadrunner hereafter) to challenge and assess the bimanual speed and precision of the child.

The child has the task to pursue a white spot (hereafter referred to as *walker*) that moves over the surface of the cube. The player has to make sure that the walker is always on top of the cube.

[3] is it necessary to cite here? then ask Svetlana to provide a reference

[4] `http://www.st.com/content/st_com/en/products/mems-and-sensors/accelerometers/lis3dh.html`

[5] `http://www.compuphase.com/pawn/pawn.htm`

The walker can move one step in three different directions forward, forward right and forward left. Steps performed by the walker can only be made to adjacent cube squares. The game is based on the assumption that it will be more difficult for the child to follow the walker if the delays i.e. intervals between the steps will decrease during the game. It should be obvious that the degree of difficulty of the game increases with shorter step-change perturbations.

The game uses a series of three levels of increased difficulty (referred to as *degree of difficulty* hereafter). A game lasts for 120 seconds and therefore each degree has a duration of 40 secs. The delay between steps at degree 2 and 3 is 80% resp. 70% of the delay used at degree 1. A step in one of the three possible directions is chosen randomly using uniform probabilities.

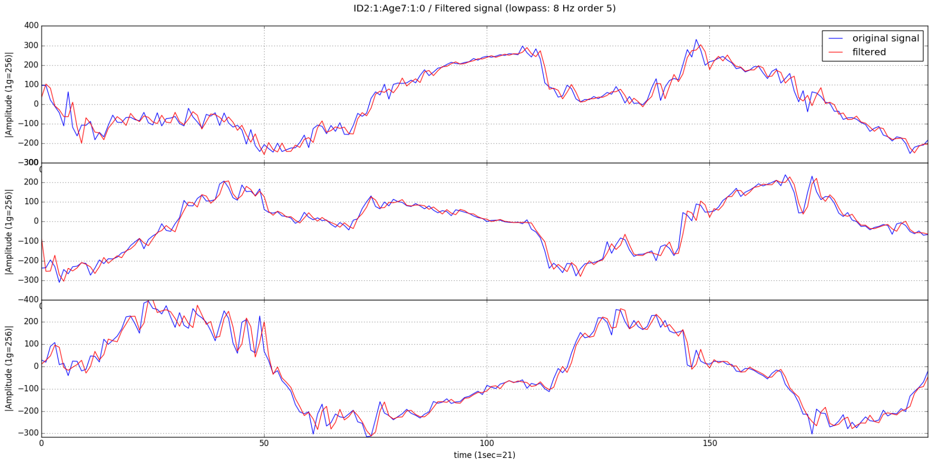
An electromyography (EMG) signal was captured for evaluating and recording the electrical activity produced by skeletal muscles of the children[6]. These measurements were not synchronized with the accelerometer data and therefore not used in the data analysis.

Although the game software provides a sampling frequency of 125 Hz the experimental setup could only capture samples with a frequency of approximately 20.8 Hz. Figure 7 illustrates the capturing of the accelerometer data during the experiments before they are being analyzed.

3 Data Analysis

The accelerometer data was digitally low-pass filtered with a cutoff frequency of 8 Hz to remove high frequency noise. [18] showed that few observable harmonics of volitional movement exist beyond 10 Hz. An example of a filtered 3-axial accelerometer signal is shown in figure 2.

Figure 2: Example of 8 Hz low-pass filtered accelerometer signal



For the further analysis the signal magnitude was calculated for each sample. The term *Movement Intensity* (MI) was coined in [5] and will be used here instead of signal magnitude. The MI was calculated as follows, $MI(t) = \sqrt{a_x(t)^2 + a_y(t)^2 + a_z(t)^2}$ with $a_x(t)$, $a_y(t)$ and $a_z(t)$ being the acceleration measured on the x-axis, y- and z-axis, respectively at time t. The movement intensity is invariant to the the sensor's orientation, which provided robustness against sensor displacement. All mathematical calculations in this study were performed using Python 2.7 with Numpy module 1.11.1.

For each child the accelerometer data was first segmented into 3 epochs of 40 seconds that coincide with the 3 degrees of difficulty of the game, before the features were extracted. Figure 6 visualizes the pre-processing pipeline of the accelerometer data.

[6] children were wearing an electromyograph on the upper right or left arm.

3.1 Feature extraction

Time domain

Statistical & envelope metrics

Simple statistical and envelope metrics (e.g. minimum, range and mean) can be used to extract basic signal information from raw sensor data. These techniques were often used in practical activity recognition algorithms [6] as well as in motor skill assessment experiments [12, 13]. A list of all the features that were calculated in the time domain are shown in Table 2.

The mean movement intensity is frequently used as a feature in rehabilitation [11] and activity recognition [19]. In clinically environments, it is related to translational movements in space and can be used to detect postural transitions [5].

Common measures of smoothness of movement are jerk measures [7] and spectrum-based measures [8] (e.g. power spectrum entropy). This study calculated both measures and used a feature ranking algorithm to obtain their predictive power. Based on the work of [7] the mean squared jerk (MSJ) was calculated which is a dimensionless jerk measure

$$MSJ = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \left(\frac{da}{dt}\right)^2 dt$$

where $\frac{da}{dt}$ denotes the time derivative of the movement intensity and t_1 and t_2 denote the begin respectively end of the time interval over which the mean was calculated. A simple 1D derivative filter was used to compute the change of movement intensity between successive samples.

Table 2: Features of time domain

No.	Feature
1	Minimum MI
2	Maximum MI
3	Mean MI
4	MI variation (standard deviation)
5	Median MI
6	Range of MI
7	Root mean squared MI
8	Mean squared jerk (smoothness)

Frequency domain

Before calculating the frequency domain features a Hamming window function was applied to the filtered magnitude signal in order to alleviate spectral leakage. The Fourier Transform (FT) of the signal was calculated with a fast Fourier Transform (FFT) algorithm. The frequency spectrum reflects the frequencies at which the movement was performed. Table 3 shows the list of features that were calculated in the frequency domain.

DC component

The DC component is the first coefficient in the spectral representation of a signal and its value is often much larger than the remaining spectral coefficients. It is equal to the average value of the signal and used as signal characteristic in several activity recognition approaches [19].

Dominant frequency

The dominant frequency of the performed motor task was calculated as the frequency associated with the highest energy of the FT acceleration signal (excluding the DC component).

Spectral energy

The spectral energy of the signal can be computed as the squared sum of its spectral coefficients normalized by the length of the sample window (i.e. the length of the epoch). The DC component is generally excluded from this calculation. The formula for the calculation of the spectral energy is outlined here.

Table 3: Features of frequency domain

No.	Feature
9	DC component
10	Dominant frequency
11	Spectral Energy
12	Power spectral entropy

Power spectral entropy (PSE)

The information entropy of the power spectrum is called power spectral entropy. For example [19] have used the PSE to distinguish between activities with similar energy levels. The algorithm can be summarized as:

The random variable X represents the states of an uncertain system, the value of X is equal to $\{x_1, x_2, \dots x_n\}$ and $n \geq 1$. The corresponding probability is $P = \{p_1, p_2, \dots p_N\}$ and $0 \leq p_i \leq 1, i = 1, 2, \dots N$ under the constraint that $\sum_{i=1}^{N} p_i = 1$.

1. The Discrete Fourier Transform $X(\omega_i)$ of a signal can be obtained by FFT where ω_i is the frequency point of the number i;
2. Calculate the power spectral density (PSD)
$$\hat{P}(\omega_i) = \frac{1}{N} |X(\omega_i)|^2$$
3. Normalize $\hat{P}(\omega_i)$, and obtain the PSD distribution function
$$p_i = \frac{\hat{P}(\omega_i)}{\sum_i \hat{P}(\omega_i)}$$
4. Then substitute p_i into the formula for the information entropy of the system
$$H = -\sum_{i=1}^{N} p_i \log p_i$$

Game state features

Degree of difficulty

As mentioned above the roadrunner game was composed of 3 degrees of difficulty. The degree d with $d \in \{1, 2, 3\}$ was captured together with all other features in order to use this information in the analysis (e.g. to detect the step-changes).

Note, this feature was not captured explicitly in the current work. As mentioned before, the game was segmented in fixed epochs of 40 secs with increased degree of difficulty. Therefore the step-changes could be derived afterwards in the analysis. In future experiments the degree of difficulty of an epoch will be a randomized property and needs to be collected explicitly during the experiment.

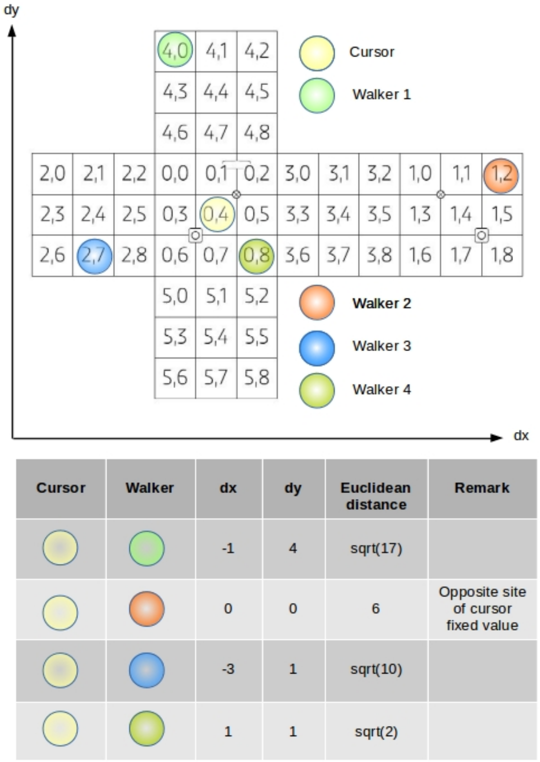
Error measure

In order to measure how accurately a child can pursue the optimal trail of the walker, an error measure was defined. Figure 3 visualizes the unfolded cube and illustrates four possible situations that exemplifies the calculation of this error measure. The so called *cursor* (the yellow circle at position (0,4)) is defined as the gravity cursor that indicates the top of the cube and is calculated by means of the acceleration data provided by the system[7]. As explained before the goal of the game is to keep the walker exactly on the same position (i.e. square) as the gravity cursor.

The software of the cube provides a function that calculates the distance of the cursor and the walker in the xy-plane at time t denoted as $\Delta x(t)$ and $\Delta y(t)$ where $\Delta x, \Delta y \in \mathbb{Z}$.

[7] this location was not visible for the player.

Figure 3: Visualizing the calculation of the error measure



Cursor	Walker	dx	dy	Euclidean distance	Remark
●	●	-1	4	sqrt(17)	
●	●	0	0	6	Opposite site of cursor fixed value
●	●	-3	1	sqrt(10)	
●	●	1	1	sqrt(2)	

The experimental data contained the sum $d(t) = \Delta x(t) + \Delta y(t)$ for each timestamp t. Obviously this should have been the sum of the absolute values $|\Delta x(t)| + |\Delta y(t)|$. Besides, after the experiments it was discovered that the cube software assigns a value of zero to both measures in case the cursor and the walker are located on opposite sides of the cube. Therefore the data w.r.t. the error measure was expected to be unreliable and useless for the analysis.

Note, the upcoming experiments will capture the separate $\Delta x(t)$ and $\Delta y(t)$ values. As a final error measure the Euclidean distance will be calculated between the cursor and the walker in $\mathbb{R}^2$.

$$d(t) = \sqrt{|\Delta x(t)|^2 + |\Delta y(t)|^2}$$

Table 4: Game state features

No.	Feature
13	Error measure
14	Maximum overshoot error
15	Error correction measure

A fixed error value of 6 will be assigned to all situations in which the walker and the cursor are located on opposites sides of the cube. Naturally the measure is equal to zero in case the cursor and walker are exactly positioned on the same location. The measure is maximally in case both objects are positioned on opposite sites of the cube.

Error correction due to step-change perturbations

Note, the features described in this section were not calculated in the current work. They are based on the error measure outlined in the previous section. These features are intended to be used in future work.

In [20] Bavassi et al. used among others small step-change perturbations in a finger-tapping task that revealed inherent nonlinearities in the underlying error correction mechanism. The roadrunner game with its transition points between game levels can be interpreted as step-change perturbations to which the child will react with a sensorimotor synchronization response in order to align her movements to the altered speed of the game.

Research has shown [20, 21] that human motor response to a step-change exhibit considerable overshoot[8] before approaching the new baseline. Another interesting finding from the step-change perturbations is that the overshoot is only displayed for positive perturbations, i.e., when the period of the sequence is increased but not when it is decreased [21].

Based on these findings the following 2 features were designed

1. *Maximum overshoot error (MOE)*, maximum error in the 100ms interval following the step-change perturbation where $MOE = \max\left[d(t_0), d(t_1)\right]$ and $\Delta t = t_1 - t_0 \approx$ 100ms;
2. *Error correction measure (ECM)*, sum of errors until the new baseline (error) is achieved, $ECM = \int_{t_0}^{t_1} d(t)dt$ and $\Delta t = t_1 - t_0 \approx$ 2000ms[9].

The design of these features hinge on the assumption that children with delayed fine motor skill development will react less optimal i.e. slower to the step-change perturbations than children without fine motor skill deficiencies and will therefore exhibit significantly larger overshoots and error correction values [10].

Table 4 lists all game state features.

Feature selection and classification

In order to reduce the dimensionality of the data ReliefF [9] was used as a feature selection algorithm. The fundamental idea behind the algorithm is to select features that can maximally distinguish between classes.

The ReliefF algorithm which is an extension of the original Relief algorithm proposed by [10], has been designed for multiclass problems and is based on the k nearest neighbors from the same class, and the same number of vectors from different classes. It is more robust in the presence of noise in the data. Relief algorithms represent an original approach to feature selection, that is not based on the evaluation of one-dimensional probability distributions [9]. Finding nearest neighbors assures that the feature weights are context sensitive, but are still global indices (see also [22] for another algorithm of the same type).

In the first step the optimal feature ranking was determined by the ReliefF algorithm. A backward feature selection process was performed in the second step by training, validating and testing 4 different classification algorithms and using the accuracy percentage as a measure to assess the discriminatory ability of the N_i ranked features in subset $\mathcal{F}_{N_i}$ where $N_i \in \{N, N-1\ldots, 1\}$ denotes the number of features in a subset (with $i \in \{1, \ldots, N\}$) and N denotes the total number of ranked features. The subset $\mathcal{F}_N$ contained all N top ranked features and each successive set contained the top N_i ranked features where the lowest ranked feature from the previous set was omitted. The following classifiers where used

1. Support Vector Machines (SVM);
2. Random Forest (RF);
3. Extreme Gradient Boosting machine (XGB);
4. Gaussian Naive Bayes (GNB).

[8] to be explained

[9] validate and discuss with Antoine which time interval would be appropriate

[10] tiny synchronization errors or small differences between the interstimulus interval and the interresponse interval could rapidly accumulate and make the responses drift away from the stimuli, as it is that children with serious fine motor skill deficiencies could possibly not adjust to the perturbations

The accuracy percentage was derived for each classifier by performing a four-fold cross-validation. The analysis was implemented using the machine learning toolbox *scikit-learn* release 0.18 and the ReliefF implementation from [23].

4 Results

This section presents the preliminary results of the experiments and data analysis outlined above. Results are described for 2 scenarios. In scenario (1) the game state feature *error measure* was excluded in the analysis whereas in scenario (2) this feature was a member of the feature set to be evaluated. As illustrated earlier the error measure was supposed to be unreliable due to the way it was captured.

Scenario (1) contained 12 features and Table 7 shows their rank determined by the ReliefF algorithm. The power spectral energy which is a measure of randomness is the best ranked feature followed by the spectral energy and the DC component. The best ranked time domain feature is the mean squared jerk which is a measure of movement smoothness. It is subsequently pursued by the root mean squared movement intensity value which is a measure for the dynamic energy of a movement [12].

Using the ranked feature list of scenario (1) as input for the backward feature selection process that uses the accuracy scores of the four mentioned classifiers as evaluation measure, the results indicate that the Gaussian Naive Bayes classifier with an accuracy score of 77% and a standard deviation of 0.185 attains the best performance on the set of 5 best ranked features shown in table 7.

The other 3 classifiers obtain their best discriminatory performance equal to 70% with a subset of 4 top ranked features as can be seen in table 5.

Scenario (2) included as 13^{th} feature the error measure which quantifies how accurately a child can perform the predefined task. In the feature ranking obtained with the ReliefF algorithm this feature claims the highest rank. The results of the feature ranking and backward feature selection process for scenario (2) are shown in table 6 resp. figure 8.

Table 5: Top accuracy scores of classifiers during backward feature selection - scenario (1)

Rank	Classifier	Max accuracy [%]	2σ	# of top ranked features
1	GNB	77	0.37	5
2	RF	70	0.38	4
2	SVM	70	0.38	4
2	XGB	70	0.38	4

Table 6: Top accuracy scores of classifiers during backward feature selection - scenario (2)

Rank	Classifier	Max accuracy [%]	2σ	# of top ranked features
1	RF	77	0.33	8
1	SVM	77	0.33	8
1	XGB	77	0.33	8
2	GNB	73	0.36	6

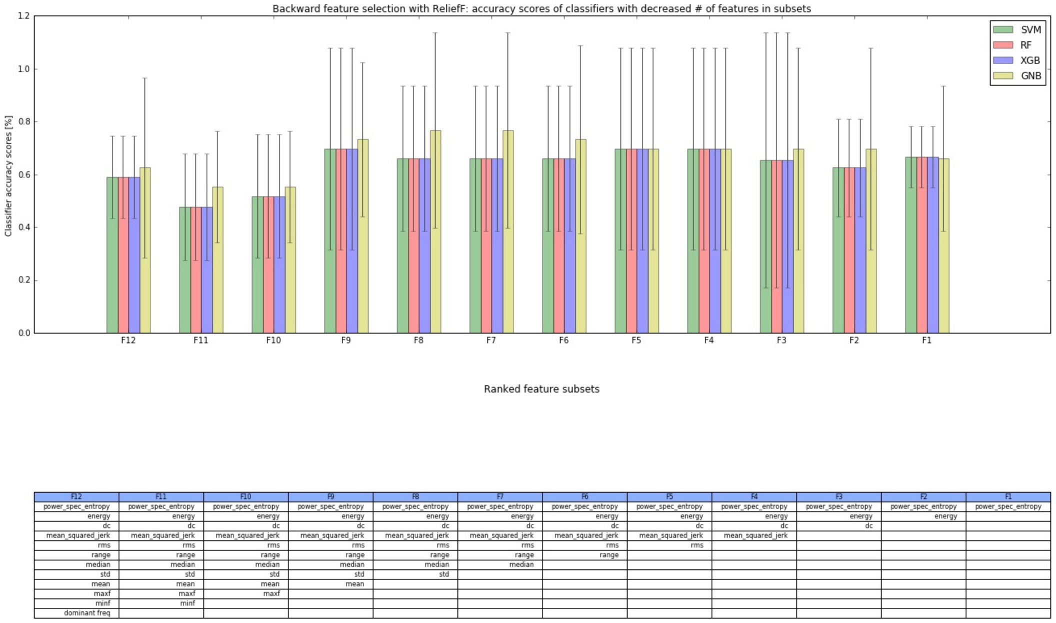
Table 7: Top ranked features by the ReliefF algorithm - scenario (1)

Rank	Feature
1	Power spectral entropy
2	Spectral energy
3	DC component
4	Mean squared jerk
5	RMS
6	Range
7	Median
8	Standard deviation
9	Mean
10	Max
11	Min
12	Dominant frequency

Table 8: Top ranked features by the ReliefF algorithm - scenario (2)

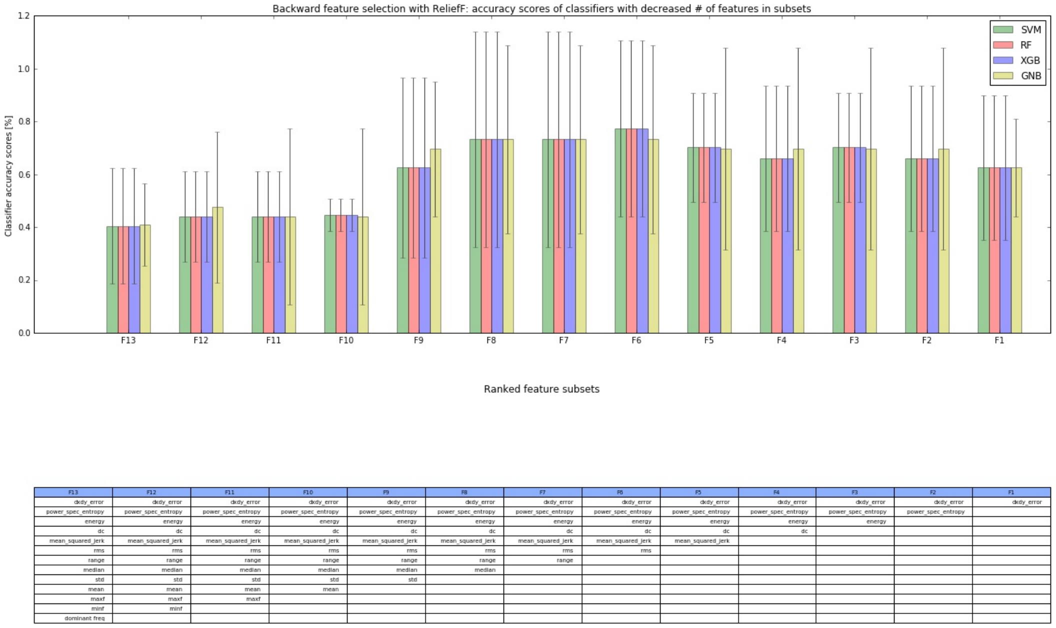
Rank	Feature
1	Error measure
2	Power spectral entropy
3	Spectral energy
4	DC component
5	Mean squared jerk
6	RMS
7	Range
8	Median
9	Standard deviation
10	Mean
11	Max
12	Min
13	Dominant frequency

Figure 4: Results backward feature selection process (error bars represent 2σ) - scenario (1)



F12	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1
power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy
energy	energy	energy	energy	energy	energy	energy	energy	energy	energy	energy	
dc	dc	dc	dc	dc	dc	dc	dc	dc	dc		
mean_squared_jerk	mean_squared_jerk	mean_squared_jerk	mean_squared_jerk	mean_squared_jerk	mean_squared_jerk	mean_squared_jerk	mean_squared_jerk	mean_squared_jerk			
rms	rms	rms	rms	rms	rms	rms	rms				
range	range	range	range	range	range	range					
median	median	median	median	median	median						
std	std	std	std	std							
mean	mean	mean	mean								
maxf	maxf	maxf									
minf	minf										
dominant freq											

Figure 5: Results backward feature selection process (error bars represent 2σ) - scenario (2)



F13	F12	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1
dxdy_error	dxdy_error	dxdy_error	dxdy_error	dxdy_error	dxdy_error	dxdy_error	dxdy_error	dxdy_error	dxdy_error	dxdy_error	dxdy_error	dxdy_error
power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy	power_spec_entropy	
energy	energy	energy	energy	energy	energy	energy	energy	energy	energy	energy		
dc	dc	dc	dc	dc	dc	dc	dc	dc	dc			
mean_squared_jerk	mean_squared_jerk	mean_squared_jerk	mean_squared_jerk	mean_squared_jerk	mean_squared_jerk	mean_squared_jerk	mean_squared_jerk	mean_squared_jerk				
rms	rms	rms	rms	rms	rms	rms	rms					
range	range	range	range	range	range	range						
median	median	median	median	median	median							
std	std	std	std	std								
mean	mean	mean	mean									
maxf	maxf	maxf										
minf	minf											
dominant freq												

5 Discussion

References

[1] Hildebrandt, V.H., A.M.J. Chorus, J.H. Stubbe, Trendrapport Bewegen en Gezondheid 2008/2009. 2010, TNO Kwaliteit van Leven: Leiden.

[2] Kurtz, L.A., Understanding motor skills in children with dyspraxia, ADHD, autism, and other learning disabilities. 2008, London: Jessica Kingsley Publishers.

[3] Fisher, A, et al., Fundamental movement skills and habitual physical activity in young children. Medicine and Science in Sports and Exercise, 2005. 37: p. 684 - 688.

[4] Cliff, DP, AD Okely, LM Smith, K McKeen, Relationships between fundamental movement skills and objectively measured physical activity in preschool children. Pediatr Exerc Sci, 2009. 21: p. 436 - 449.

[5] Strohrmann, Christina, et al. "Monitoring motor capacity changes of children during rehabilitation using body-worn sensors." Journal of neuroengineering and rehabilitation 10.1 (2013): 1.

[6] Kern N, Schiele B, Schmidt A (2007) Recognizing context for annotating a live life recording. Personal and Ubiquitous Computing 11(4):251–263

[7] Hogan N, Sternad D: Sensitivity of smoothness measures to movement duration, amplitude, and arrests. J Motor Behav 2009, 41(6):529–534.

[8] Kojima M, Obuchi S, Mizuno K, Henmi O, Ikeda N: Power spectrum entropy of acceleration time-series during movement as an indicator of smoothness of movement. J Physiol Anthropol 2008, 27(4):193–200.

[9] M. Robnik-Šikonja and I. Kononenko, "Theoretical and Empirical Analysis of ReliefF and RReliefF," Machine Learning, vol. 53, 2003, pp. 23 - 69.

[10] K. Kira and L.A. Rendell, "A practical approach to feature selection," Ninth International Workshop on Machine learning, Aberdeen, Scotland: Morgan Kaufmann, 1992, pp. 249 - 256.

[11] Bonato P: Advances in wearable technology and applications in physical medicine and rehabilitation. J NeuroEng Rehabil 2005, 2:2.

[12] Patel S, Hughes R, Hester T, Stein J, Akay M, Dy J, Bonato P: Tracking motor recovery in stroke survivors undergoing rehabilitation using wearable technology. In Proceedings of the 32nd Annual International Conference of the IEEE EMBS: IEEE; 2010:6858–6861.

[13] Patel S, Hughes R, Hester T, Stein J, Akay M, Dy J, Bonato P: A novel approach to monitor rehabilitation outcomes in stroke survivors using wearable technology. Proc IEEE 2010, 98:450–461.

[14] Hester T, Hughes R, Sherrill D, Knorr B, Akay M, Stein J, Bonato P: Using wearable sensors to measure motor abilities following stroke. In Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on: IEEE Computer Society; 2006: 4, pp. 5–8.

[15] Del Din S, Patel S, Cobelli C, Bonato P: Estimating fugl-meyer clinical scores in stroke survivors using wearable sensors. In Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE: IEEE; 2011:5839–5842.

[16] Parnandi A, Wade E, Mataric M: Motor function assessment using wearable inertial sensors. In Engineering in Medicine and Biology Society (EMBC), Annual International Conference on: IEEE; 2010:86–89.

[17] Zhang M, Lange B, Chang CY, Sawchuk AA, Rizzo AA: Beyond the standard clinical rating scales: fine-grained assessment of post-stroke motor functionality using wearable inertial sensors. In International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC); 2012.

[18] Hotraphinyo LF, Riviere CN. Three-dimensional accuracy assessment of eye surgeons. Proceedings of the 23rd annual international conference of the IEEE Engineering in Medicine and Biology Society, Vols 1-4. 2001;23:3458-3461.

[19] Bao, Ling. Physical activity recognition from acceleration data under semi-naturalistic conditions. Diss. Massachusetts Institute of Technology, 2003.

[20] Bavassi, M. Luz, Enzo Tagliazucchi, and Rodrigo Laje. "Small perturbations in a finger-tapping task reveal inherent nonlinearities of the underlying error correction mechanism." Human movement science 32.1 (2013): 21-47.

[21] Thaut, M. H., Miller, R. A., Schauer, L. M. (1998). Multiple synchronization strategies in rhythmic sensorimotor tasks: phase vs period correction. Biological Cybernetics, 79, 241–250.

[22] Yang, Yiming, and Jan O. Pedersen. "A comparative study on feature selection in text categorization." ICML. Vol. 97. 1997.

[23] Randy Olson, (2016). ReliefF: First release [Data set]. Zenodo. http://doi.org/10.5281/zenodo.47803

6 Appendix A

6.1 Flowchart of pre-processing steps

Please see figure 6 for a visualization of the pre-processing flow.

Figure 6: Pre-processing flowchart

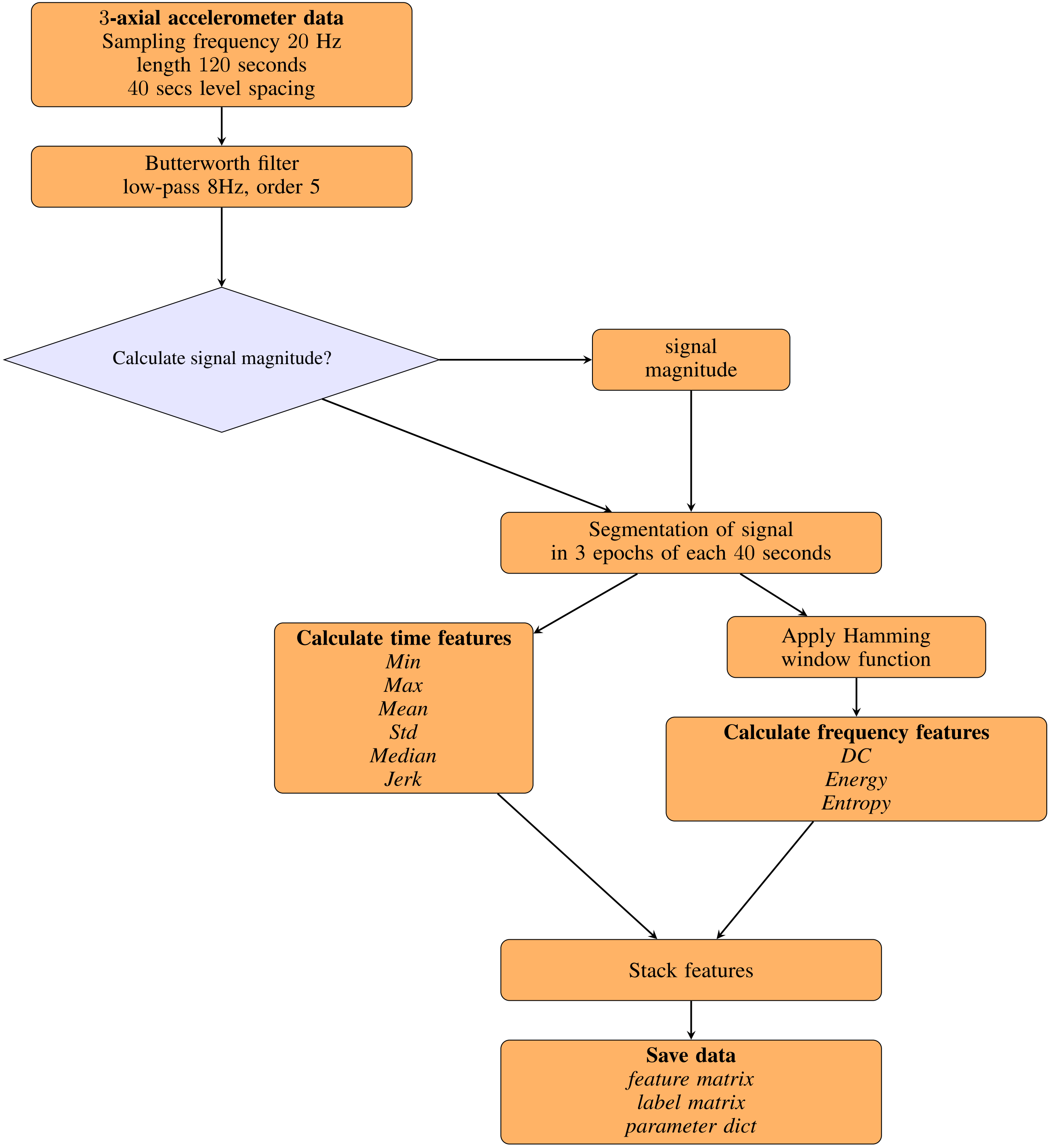


Figure 7: Flowchart of capturing accelerometer measurements

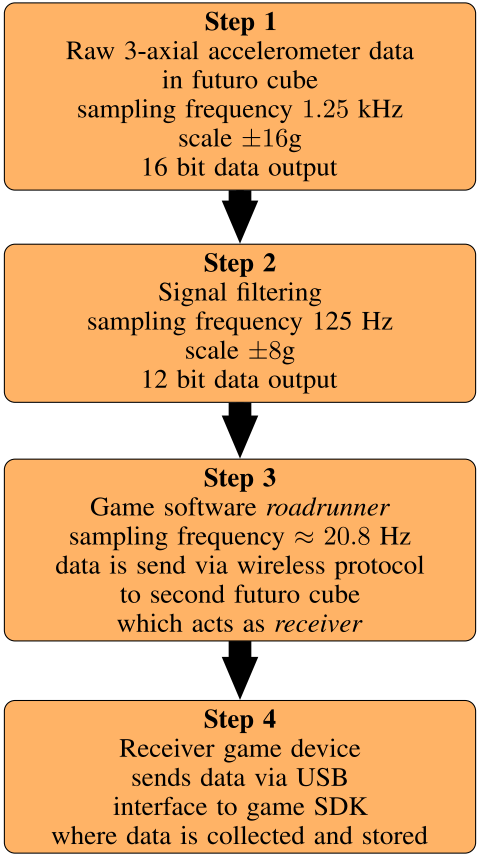


Figure 8: Example 1, LED indicating the cursor position on the cube

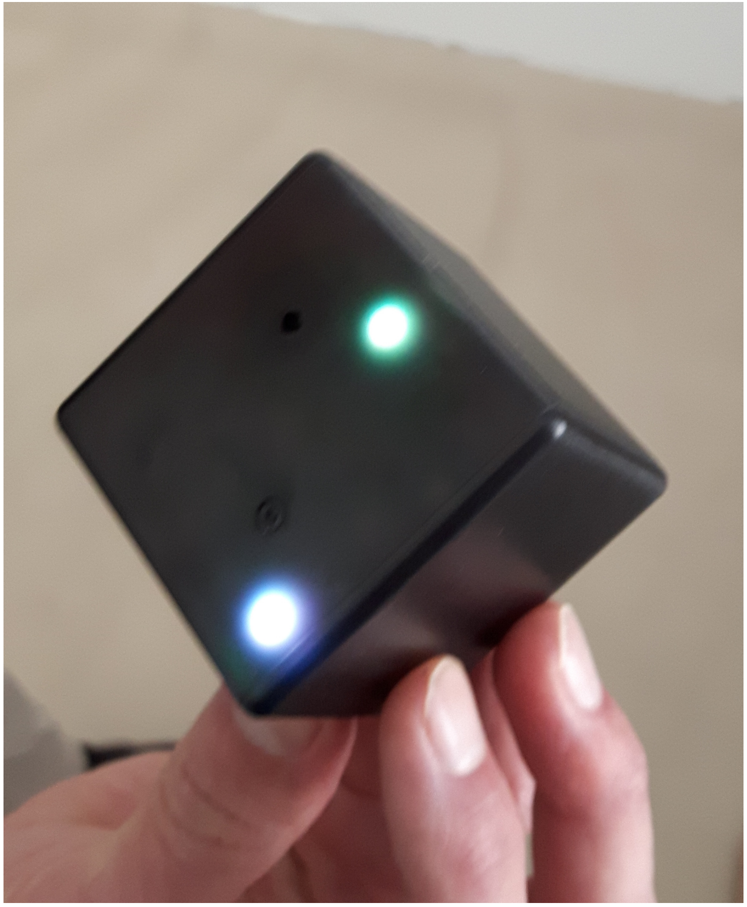


Figure 9: Example 2, LED indicating the cursor position on the cube

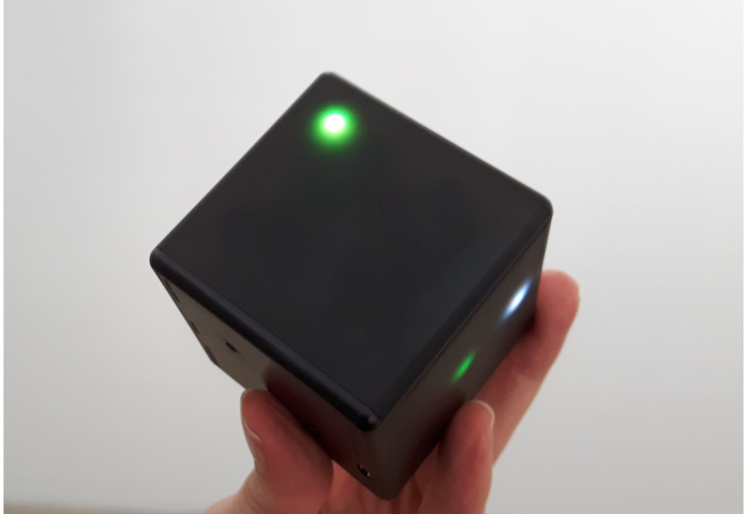


Figure 10: Upward position based on accelerometer