**Title**: Fall risk at a festival

**Client**: BRAVO project @HvA (Amsterdam University of Applied Sciences)

**Contact**: Saskia Robben [s.m.b.robben@hva.nl](mailto:s.m.b.robben@hva.nl)

**Date/version**: 2018-11-26

**Backgroud**

Falling is a major problem: In the Netherlands, every 5 minutes an older adult (65+) is treated at the hospital after a fall. On a daily basis, 11 persons die from falling [1]. Early identification of elderly with a high risk of falling is a key factor in preventing fall accidents. Current screening tools are either not discriminative, costly, time-consuming, and/or require professional expertise.

Goal of the BRAVO project is to develop a pre-screening tool that is accessible and easy to use, so it can be used on a large scale. The dataset used for this assignment is collected in an pilot on the Lowlands Festival [2].

Physiological impairments affect normal movement patterns (Fig 1). Therefore, gait analysis is a promising non-invasive method for identification of physiological fall risk factors. This pilot investigates the feasibility of using a short test protocol, with a low-cost optical 3D sensor, to categorize gait patterns. As data quality and quantity is insufficient for conventional biomechanical gait analysis, a machine-learning approach should be used to analyze 3D time series. In this study [3], a controlled intervention was used with healthy young adults to induce an altered gait pattern to test this proof of concept.

**Methods**

* 205 participants
* The test protocol included rising from a chair, 4x3m walking at comfortable speed and sitting down. This was repeated in two conditions:
  + Control
  + Glasses: Distorted depth vision (induced by simulation goggles)
* Sensor:
  + Microsoft Xbox One Kinect
* Preprocessing:
  + Remove erroneous data
  + Segmentation of gait sections
  + Low-pass filtering
  + 3d data of 1 joint (mid-spine) selected.
  + (See fig 2 for example projections of the resulting data)

**Goal/Assignment**

When depth vision is distorted, most people walk differently. As humans we can clearly see this, but the question is whether we can automatically differentiate between these two types of walks.

*Task*: make a classifier that can differentiate between the two conditions (control vs. glasses)

*Preferred programming language*: Python (or R)

**Bonus problems**

* Use different labels
  + Classify no/low or high alcohol intake
    - Similar as main goal, only different labels.
  + Predict continuous alcohol score (regression model)
  + Use balance score (labels are less reliable because data was obtained from a suboptimal balance test)
    - Is it possible to detect balance from walks?
    - With this problem: only use the walks without glasses, as the glasses do distort the walks but don't distort the balance test
  + Simulate “real world” problem by using only one condition for each person (select randomly)
* Use all the other data (joints) (not yet preprocessed)
* Use a related dataset (older adults performing a similar task) (not yet preprocessed)

**References and additional reading:**

* [1] Factsheet Letsel door Valongevallen 65+ in 2017, [www.veiligheid.nl/valpreventie/](http://www.veiligheid.nl/valpreventie/)
* [2] [www.lowlands.nl/english/](http://www.lowlands.nl/english/)
* [3] Aukje de Vrijer & Saskia Robben (2018) Towards an accessible pre-screening tool for fall risk assessment: Automated gait analysis using a machine-learning approach. EU Falls Festival 2018, Manchester. (*poster*)

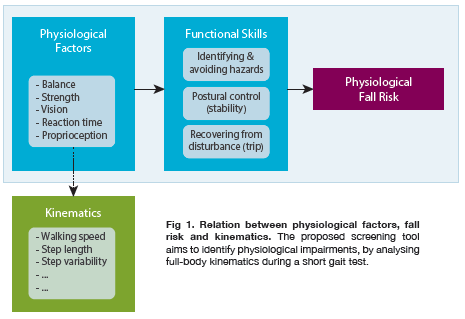
**Acknowledgements**

This research was financially supported by SIA. We thank our partners from the BRAVO-project, colleagues, students and the organization of Lowland Science for their help and commitment!

**During kick-off**

* Additional material: movie/clips and pictures to clarify setup
* Opportunity to try glasses that distort depth perception
* Q&A on problem & data
* Agreement on additional contact/feedback moments

**Figures:**



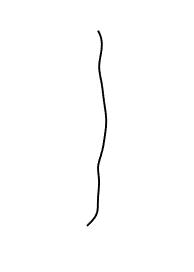
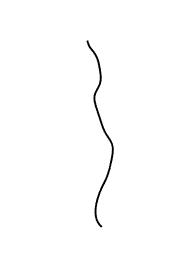
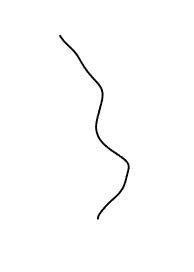
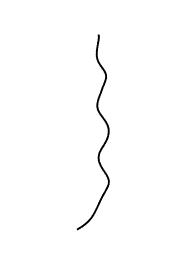
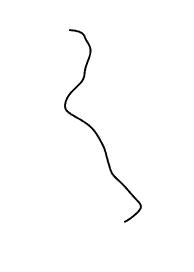


Fig 2: example walks (2d-projection of spine mid point)

**Description of the data:**

|  |  |
| --- | --- |
| **Row** | **Description** |
| TestID | Unique ID of Test; For example "169B03" is subject ID 169; condition B (with glasses distorting vision); third walk (approx 3 walks per person) |
| Conditie | With or without distorted depth perception |
| SubjectID | Unique ID of Participant |
| TrialNr | N-th time the participant performed the test |
| NumberOfSegments | Total number of gait segments in this test |
| SegmentNr | N-th gait segment of the test |
| WalkingDirection | Whether subject is walking towards or away from the sensor |
| SegmentData | Contains the actual xyz data of the spine-mid point (see next page) |
| Alcohol | Breath alcohol concentration; (e.g. use with a cut-off point of 0.5). Note there is a disbalance in the distribution of alcohol intake in the dataset |
| Height | Subject height |
| Weight | Subject weight |
| Age | Subject age |
| BMI | Body Mass Index |
| Balance\_MLrange | Balance metric |
| Balance\_MLstdev | Balance metric |
| Balance\_MLmeanVelocity | Balance metric |
| Balance\_APrange | Balance metric |
| Balance\_APstdev | Balance metric |
| Balance\_APmeanVelocity | Balance metric |
| Balance\_MeanVelocity | *Preferred* balance metric (balance metrics are all calculated from the data obtained in a short balance test with visual feedback) |
| Gait\_Velocity | Median gait velocity of head during gait segment in Z-direction |
| Movement\_velocity | Median velocity of head during gait segment in the horizontal plane |

**Python example code to help loading the matlab-datafile**

*# Load the matlab file*

from scipy.io import loadmat

matdata = loadmat("../Data/SpineMidData\_21032018.mat")

*# Remove some irrelevant matlab metadata*

spineMidData=matdata['SpineMidData']

*# Inspect the headers:*

spineMidData[:1]

*# Get a column of data*

spineMidData[:,0]

*# SegmentData*

*# contains the actual xyz data of the spine-mid point. In the jpg's you see an example of a 2d projection of such walks*

*#note: number of frames is not the same for all walks*

*#note2: similar data exists for all joints that software for kinect stored. This data is not yet preprocessed.*

spineMidData[:,7]