Fall simulator for supporting supervised Machine Learning techniques in wearable devices

Armando Collado-Villaverde Mario Cobos Pablo Muñoz Maria D. R-Moreno

Intelligent Systems Group - Computer Engineering Department

August, 2020

Outline

Introduction

Motivation

Fall detection is a significant issue in elderly care

- Direct trauma consequences
- Consequence of heart conditions and can originate new others

Related work with several approaches

- Image and Video processing
- Dedicated devices
- Audio detection
- Smartwatch

Last one is used by the LARES project

- Passive teleassistance system
- Network of wearables and IoT

Introduction

Simulator

- The main problem of fall detection in Machine Learning is the scarce amount of recorded falls
  - Involving people falling numerous times implies health risks
  - Using dummies is expensive in terms of time an required personnel
- We propose the usage of a video game engine with realistic physics to simulate falls.
- Unsupervised, capable to create a large amount of simulated, yet realistic data
- The engine allows the customization of the body dynamics.

Fall Simulator

Types of falls

We simulate the two most common types of falls that occur at home:

- **Syncope fall:** Loss of conscience without using the limbs to control the fall
  Vertical motion on two stages:
  - The knees impact the ground
  - Trunk moves forward
- **Forward fall:** Trip while walking, losing balance and falling over
  The trunk moves forward while the arms and hands cushion the fall

In our previous work we recreated the syncope falls using a dummy and the forward ones with real people.

Fall Simulator

Physics

- Physics are implemented using classical mechanics
- The accelerometer is approximated as a discrete particle in space
- The underlying engine operates using position, so the following relations are used to inform the process of extracting acceleration data:

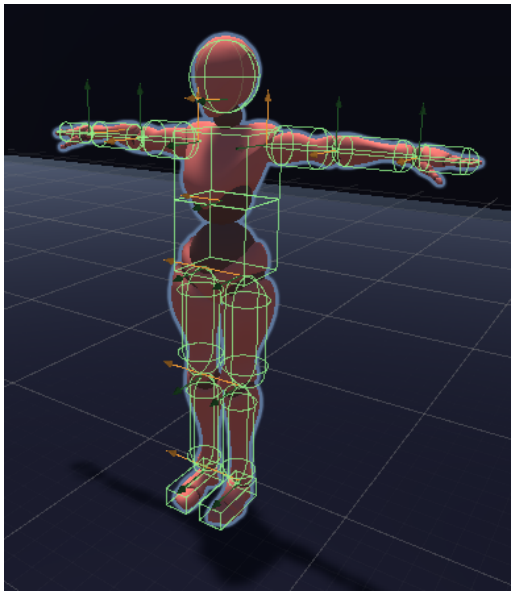
$$\vec{V}(t) = \frac{\partial \vec{P}}{\partial t}(t)$$
$$\vec{A}(t) = \frac{\partial \vec{V}}{\partial t}(t)$$

- Given the digital nature of the system, the already mentioned equations are approximated using discrete functions that encode the same ideas

Fall Simulator

Implementation

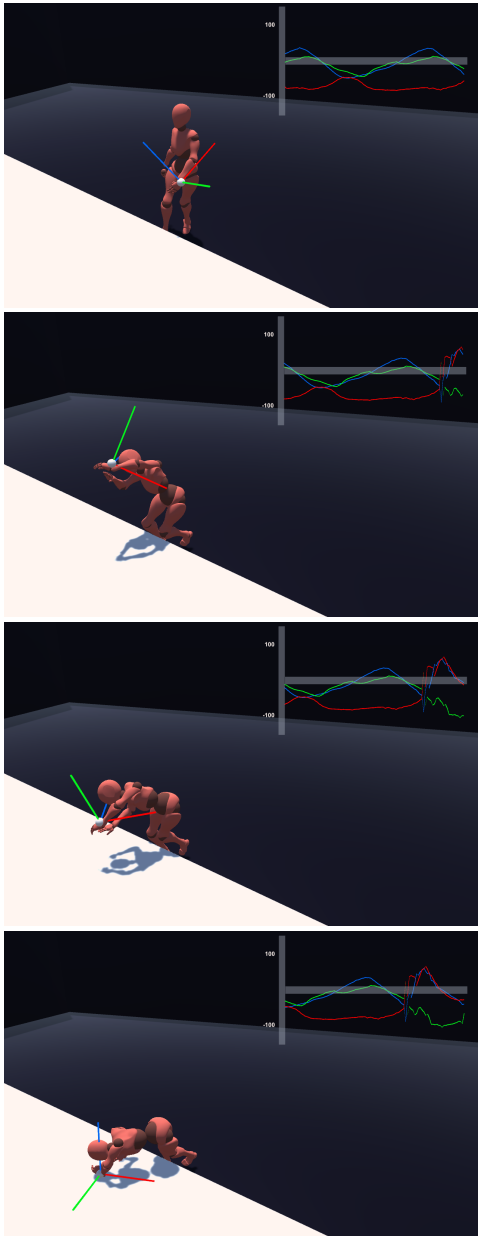
- We used Unity3d game engine which incorporates NVidia Physx for the physics calculations
- Configured a human-like 3D model with the appropriate colliders, weight and joints according to the characteristics of an elder
- To simulate both falls, we started from an animation not influenced by physics and when the fall simulation starts, the physics engine takes over
- Unity's physics engine works in a non-deterministic way, thus, each fall is different from the rest



Fall Simulator

Implementation - Forward falls

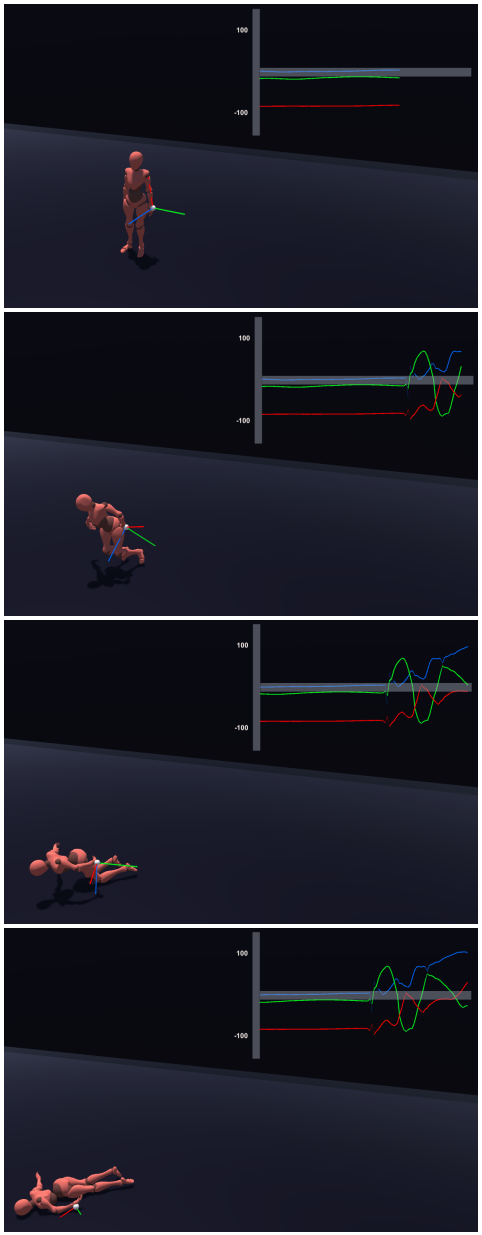
- We start playing one of several walking animations
- The humanoid collides with an obstacle in front of him, randomizing its distance, height and orientation each fall
- Then we apply forces with noise on the arms and hands according to an elder capabilities to simulate the response
- The combination of the forces, the variety of animations and positions of the obstacle guarantee a variety of falls



Fall Simulator

Implementation - Syncope falls

- We start playing one of several daily life activities
- We stop playing the animation, simulating the loss of conscience
- A randomized force is applied in a forward direction to the body, similar to the dummy recreation



Fall Simulator

Data acquisition

- The simulator records position that a smartwatch would have at fixed time steps of 20ms
- Gravity is simulated as a vector pointing towards the ground in the local space of the wrist at each time step
- Calculate $\vec{v}$ representing the resulting vector between time steps form the simulator's raw position data:

$$\vec{v} = \frac{\Delta \vec{p}}{dt} \tag{1}$$

- Then, calculate the acceleration vector as the velocity difference between time steps:

$$\vec{a} = \frac{\Delta \vec{v}}{dt} \tag{2}$$

Experimental results

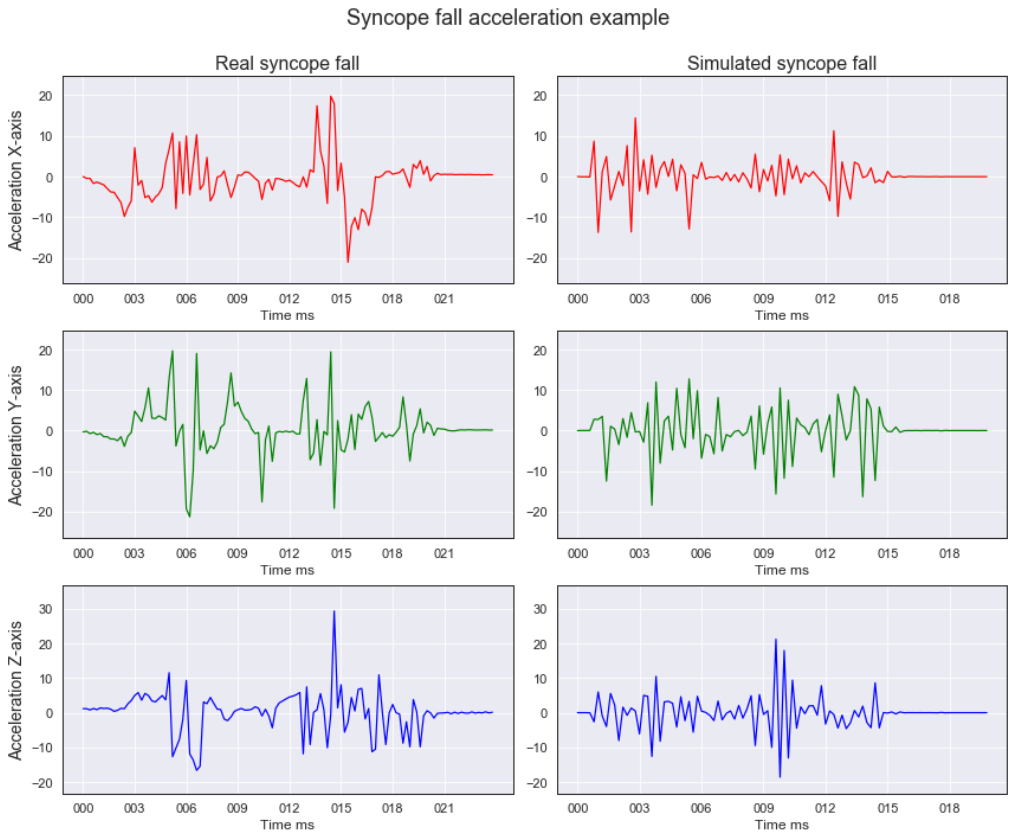
Syncope Falls

Normalized accelerometer comparison of a syncope fall

- From our experiments with dummies
- Generated in the simulator

Main differences between the data examples:

- Rigidness of the dummies causes abrupt readings
- Initial pose on the simulated falls



Experimental results

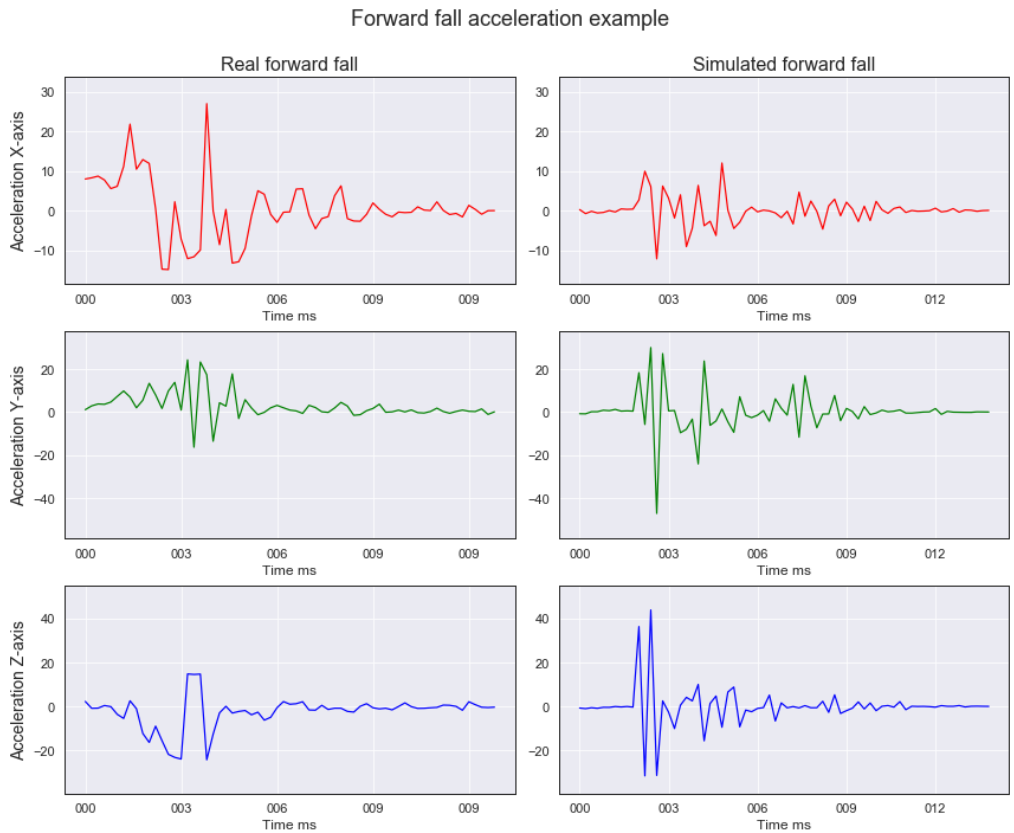
Forward Falls

Normalized accelerometer comparison of a forward fall

- Real people falling in a cushioned environment
- Generated in the simulator

Main differences between the data examples:

- Inability to fully replicate human gait, animations smooth movement
- Reactions when tripping highly tied to each subject's reactions



Experimental results

- Fall detection as a time series analysis approach present some specific problems that difficultate the comparison of the real and simulated data:
  - The real direction that each axis represents is tied to the hardware itself
  - The duration of each fall is not fixed, depends on its development
  - The starting pose plays a major role in the resulting series of vectors
- Traditional time series metrics such as correlation or dynamic time warping are not recommended, instead, ML classifiers that differentiate falls are better
- We evaluated simulated falls using the classifiers trained in our previous work using dummies for the syncope falls and real people for the forward falls and obtained 94 % accuracy using a Random Forest classifier

Conclusions

- We presented a fall simulator capable of recreating accelerometer fall samples for two types of falls
- The simulated data is suitable for Machine Learning purposes
- Future work:
  - Simulate a wider variety of falls
  - Record the accelerometer measurements in other parts of the body

Thanks for your attention!

Project available:
`https://github.com/ISG-UAH/ISG_FallSimulator`

Armando Collado Villaverde
armando.collado@uah.es