Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelor's Thesis

Matthias Lösch

Evaluation of data driven sensor fusion for activity detection and localization with accelerometers

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Matthias Lösch

Evaluation of data driven sensor fusion for activity detection and localization with accelerometers

Bachelor's Thesis eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Tiedemann
Zweitgutachter: Prof. Dr. Lehmann

Eingereicht am: 23. June 2018

Matthias Lösch

Thema der Arbeit

Evaluation of data driven sensor fusion for activity detection and localization with accelerometers

Stichworte

Multi Sensor Data Fusion, Inertialsensoren, Eingabeerkennung, IMU, Akustische Oberflächenwellen

Kurzzusammenfassung

Diese Arbeit untersucht die Verwendbarkeit von Inertialsensoren für Benutzerinteraktionen auf einer Schreibtischplatte. Kombinierte Beschleunigungs- und Drehsensordaten mehrerer inertialer Messeinheiten werden verwendet, um die Position einer Benutzereingabe zu ermitteln. Dazu wird zunächst versucht mittels Triangulation die Peilung der Quelle zu ermitteln, dann werden Machine-Learning Algorithmen angewendet.

Matthias Lösch

Title of the paper

Evaluation of data driven sensor fusion for activity detection and localization with accelerometers

Keywords

Multi sensor data fusion, inertia sensors, input detection, IMU, surface acoustic waves

Abstract

This bachelor's thesis evaluates the use of inertia sensors for turning the desktop surface into a detector for user activity or input. Combined accelerometer and gyroscope data from multiple inertia mesurement units is used to dertermine the location of the user interaction. Two approaches are evaluated, triangulation using the relative bearing of the input location relative to the sensors and machine learning strategies.

Contents

List of Tables

List of Figures

1. Introduction

1.1. Motivation

In recent years human-computer interaction has changed. While a few decades ago mechanical switches, rotary controls and sliders directly influencing an analogue electric circuit were the norm, the miniaturization and advancements in microprocessors and sensors allowed the development of a number of new input interfaces.
Touch screens and touch pads are wide spread in consumer electronics.

1.2. Approach

To denote a location on a surface, the natural human interaction is to tap there with the finger. This tapping creates vibrations which are detectable by sensors.

1.3. Theory

Tapping on a table creates acoustic waves. They propagate with the speed of sound circular from their origin. As the wave front has to travel different distances to arrive at each sensor and incoming angle of the wave front differs and the location of the origin can be inferred from these differences.

1.4. Purpose

This thesis will evaluate the whether it is possible to determine the location of finger tapping on any tabletop using consumer-grade inertia sensors. It will be evaluated whether the combination of accelerometer and gyroscope data can increase the accuracy of the calculation.

2. Related works

As shown by Braun u. a. (2015) the acoustic waves transmitted by a tabletop can be used to determine the location and even type of interaction (2.1.6). While their approach requires only a single sensor to localize the input after some calibration, it picks up only the longitudinal component of the acoustic wave.
By measuring the deflection on multiple axes geologists use different waveforms to gain more details from their only means of observation.

To try to compensate the difference in sample rate in order of two magnitudes that is found in cheap consumer grade inertial sensors, both approaches are combined. It will be evaluated if there is an obvious correlation between the readings or if machine learning strategies similar to those used by Braun u. a. (2015) can be used.

2.1. Input localization on surfaces

There are numerous existing technologies to detect the location of touch or tap input on surfaces. The most widespread and common applications are in touch screens such as smart phones, vending machines, and touch pads in laptops. In those settings the input area is defined and can be fitted with sensing technology. There are different approaches to this problem, resulting in a wide range of solutions differing in durability, sensitivity, size, energy consumption and cost. In some settings it might be desirable to turn an existing surface such as a tabletop into a touch input device, without adjusting the input area to the needs of the detection method. In this case, many properties of the material, such as stiffness and luminosity, are unknown and some kind of calibration is necessary. Colegrove (2010)

2.1.1. Electric Potential

Electric potential is the most widely used method for input localization.

Resistive

Resistive touch pads consist of two layers. If a force is applied to the upper layer it comes into contact with the lower layer and closes a circuit, allowing to measure the change in electric potential difference along the x- and y-axis of the sensor. Can be used with gloves, pens, etc., because it only requires force on the upper layer. These touch pads can be used with gloves and pens, but are not very resilient and does not react very fast.

Capacitive

Capacitive touch sensors use the electric properties of the human finger to detect changes in an electric field. A finger has a much higher electric permittivity than air, mainly due to the high water content of organic matter. This is usually archived by electrode arrays below the touch field, but in recent years the projective touch field technology also relies on measuring the change of potential difference along the two axes.

2.1.2. Optical

Infra-red Array

A very simple and cost-efficient solution is to surround the input area with an array of infra-red emitting diodes and receivers. The location of the input is directly given by the shadowed receivers on the x and y axis. (Dohse u. a., 2008, p 297)

Video Tracking

Deriving from human experience optical tracking seems to be a natural solution. Most of these systems project an infra-red grid onto the surface to determine their viewing angle. Letessier und Bérard (2004)

2.1.3. Near Field Sensors

There are several inexpensive devices promising to turn any surface into a smart surface by using ultrasound distance sensors to measure the position of fingers in front of the sensor. The area is mostly very limited. (Nonaka und Da-te, 1995, p 754)

2.1.4. Acoustic

2.1.5. Active

Surface Acoustic Wave touch panels use two receivers to measure the frequency pattern of ultrasound SAW stimulated by an emitter. (Benali-Khoudja u. a., 2004, p 6) This is very robust against humidity and like resistive touch can be operated with gloves etc. while being insensitive to water drops, making it good for wet environments.

2.1.6. Passive

Braun u. a. (2015) shows how touch/tap and even swipe input can be recognized using a single piezoelectric microphone on any tabletop surface. The frequency spectrum in the signal is analysed and machine learning is used to map the signals to positions.
There are also gadgets like Mogee Play or Pulsecontroller, which use a similar approach to convert the measured acoustic waves to input to a music synthesizer. They don't map their input signal to a defined area.

2.2. Other Applications of Acoustic Waves in Solids

2.2.1. Geology

The field of geology relies heavily on measurement of seismic waves. The low permeability of electromagnetic waves into the ground allow the use of classic image generating methods like radar only for depths up to 100m.

Seismology

In the event of an earthquake seismic waves propagate away from the epicentre in a spherical manner. Because the longitudinal waves (3.2.1) travel much faster than the transversal waves (3.2.1), they are named P-(primary) and S-waves (secondary) respectively. The distance to the epicentre can be calculated solely from the difference in arrival time without requiring a second measurement for triangulation. Their different propagation speeds cause the waves to diffract differently when the composition of the medium changes. (Anstey, 1977, p. 128) The liquid core of earth doesn't transmit S-Waves and by comparing signals from seismometers around the globe allows geologists to approximate the composition of the earth. Measuring the grounds rolling movement with gyroscopes has not been done until recently, as gyroscopes become more precise. (Suryanto, 2006, p 16)

Seismic

Similar methods are used to get a better understanding of the topmost layers of earth's crust, but the seismic event is induced artificially by detonations or simply by dropping a large mass. Geophones are used to measure the signal around the source of the waves and by interpreting reflections and refractions of the waves on different layers of stone or sediment it is possible to detect aquifers, oilfields, ore deposits or the progress of tunnel construction (Schmidt und Wuttke, cf 4.1).

2.2.2. Engineering

Ultrasonic testing is a technique to detect material deficiencies in components. A probe emits short ultrasonic impulses into the material and measures the reflections received. Deficiencies change the refractive behaviour of the ultrasonic waves and can be imaged from the recorded reflections. (Deutsch u. a., 2013, p 4)

2.2.3. Explosion Detection

Shock waves caused by explosions, for example in mines, can be detected using inertial sensors. While conventionally geophones are used to map a mine this can also be done with well aligned smartphone sensors (Thandu, 2016, cf)

3. Physical and Technical Prerequisites

The following section explains the physical principles of position detection, structure-borne sound, as well as the mechanics of inertial sensors.

3.1. Triangulation

To determine the position of point C relative to the known locations of two points A and B, a triangle with a known base can be constructed. If the angles α and β can be determined, the location of C is then known to be at the intersection of sides a and b.

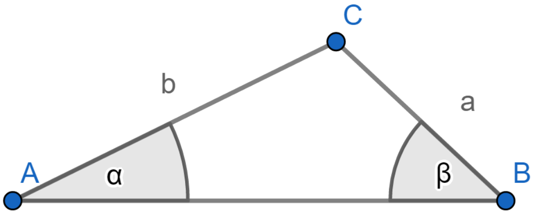

Figure 3.1.: Triangle

3.2. Acoustic Waves in Solids

Tapping onto the surface of a solid object creates pressure waves in the structure of the solid. They propagate spherically from their origin. The following section describes how waves travelling through a solid object change with distance travelled.

3.2.1. Types of Waves

(Lothar Cremer, cf)

Longitudinal Waves

Also called compression waves, these waves displace the molecules of the medium along the propagation axis of the wave. Examples are sound waves in air or the oscillation of a spring.

$$c_L = \sqrt{\frac{E}{\rho}} \tag{3.1}$$

E is the module of elasticity, also called Young's module. It is a property of the material with the density ρ

Transverse Waves

Also called shearing waves, these waves displace the medium orthogonally to the propagation axis.

$$c_T = \sqrt{\frac{G}{\rho}} \tag{3.2}$$

with G being the material-specific modulus of shear

Surface Acoustic Waves

Surface acoustic waves (SAW) occur on the surface of solids. They can be considered to extend circular from the origin and carry longitudinal (3.2.1) and transversal (3.2.1) components. They can be represented as Rayleigh-Waves, resembling the movement of the surface of liquids. Their propagation velocity c_R can be approximated with

$$c_R \approx c_T \frac{(0,874 + 1,12\mu)}{1 + \mu} \tag{3.3}$$

with Poisson's ratio μ denoting the ratio of the transversal expansion to the longitudinal compression. Lothar Cremer

3.2.2. Properties of Waves

In the following section the relevant properties of acoustic waves in solids are described

Propagation Delay

Differences in arrival time of acoustic waves are commonly used for localization (Diamant und Lampe, 2013, p 5). The difference δt_{max} is greatest when the source is located on the extension of the line between the sensors, i.e. not in the space between the sensors. By comparing any measured δt with δt_{max} the bearing of the source can be calculated. (Lothar Cremer, p 87)

Dispersion

The speed of sound in a medium depends on the wavelength and the type of the wave. Combined with the greater dissipation of higher frequency waves this leads to a widening of the signal, not only further reducing the amplitude but also increasing the wavelength λ (Anstey, 1977, p 126).

The dispersion relation describes the relationship f between the angular frequency ω and wave number k.

$$\omega = f(k) \tag{3.4}$$

As shown by (Lothar Cremer, 92,99) the phase velocity depends on the mass per distance traversed m'

$$c = \sqrt[4]{\frac{G}{m'}} \cdot \sqrt{\omega} \tag{3.5}$$

Dissipation

Dissipation or dampening is caused by friction of the molecules of the medium excited by the wave. The amplitude decreases relative to the distance that the wave has travelled.

The amplitude ψ decreases between locations $\vec{r}_0$ and $\vec{r}_d$ (Demtröder, 1998, p 393)

$$\vec{r}_0 : \qquad \psi \tag{3.6}$$

$$\vec{r}_0 + d\vec{r} : \quad \psi + d\psi = \psi + \tfrac{\delta\psi}{\delta z} d\vec{r} \tag{3.7}$$

Reflection and Refraction

Because waves are refracted or reflected on the boundaries of the medium we have to consider those as well. Due to the high speed of sound in hardwood (4.1) compared to the sample rate (4.2.1) of the sensors it can be assumed that most of the signal received will consist of reflections. Unfortunately the reflections will be scattered from all directions.

3.3. Inertial Sensors

Inertial sensors measure their displacement in reference to their own resting frame. Motion sensors (accelerometers) and rotation sensors (gyroscopes) are required to measure the six degrees of freedom when moving in three dimensional space and are often combined into one IMU (inertial measurement unit).

While IMU have been used for naval and air navigation for decades, new types of sensors are small and energy efficient enough to be present in almost every smart phone. These are based on MEMS (micro electro-mechanical system) technology. (Bernstein u. a., 2003, cf)

3.3.1. Gyroscope

A rotation sensor measures the change of rotation over time. All MEMS-gyroscopes are based on measuring the Coriolis force acting on either current or oscillations.

Draper Tuning Fork Two forks with interlocking prongs are stimulated to oscillate in opposing directions. If this structure is rotated, a Coriolis force acts on the mass of the forks and the prongs shear against each other. this results in a change of electric capacity which can be measured.Reilly u. a. (2006)

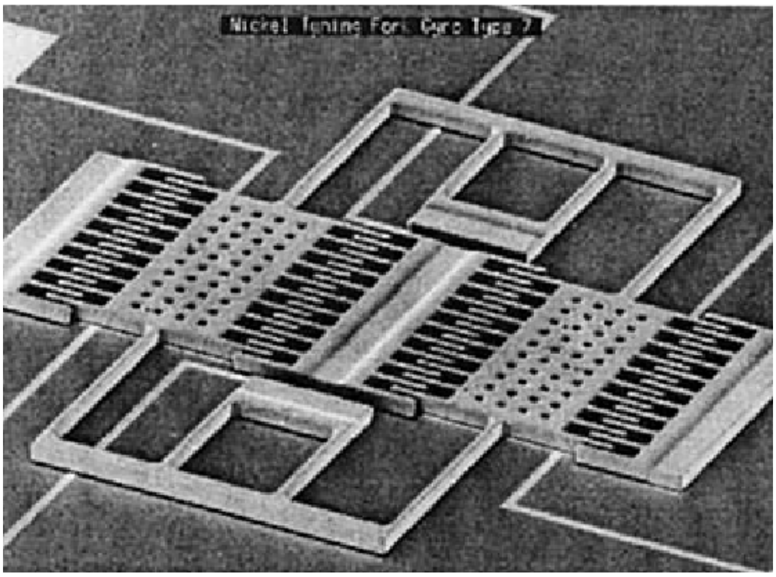

Figure 3.2.: Draper Tuning Fork
Reilly u. a. (2006)

Hemispherical Resonator Gyroscope A thin hemisphere anchored by a thick stem is stimulated to resonate. This causes the circular opening to oscillate its shape between two ellipses with orthogonal semi-axis. If the aperture is rotated in the plane of the oscillation the oscillation follows the rotation with a delay, which can be measured.

3.3.2. Accelerometer

Accelerometers measure the force acting upon them. Most accelerometers depend on measuring the force exerted by a proof mass on its attachment.

Capacitive The proof mass is acting as an electrode of a capacitor. If the container is accelerated the distance between the proof mass and the containment changes proportionally to the accelerating force acting upon it. In sensors with MEMS technology the displacement of the mass is measured using capacitors. Zhang (1998)

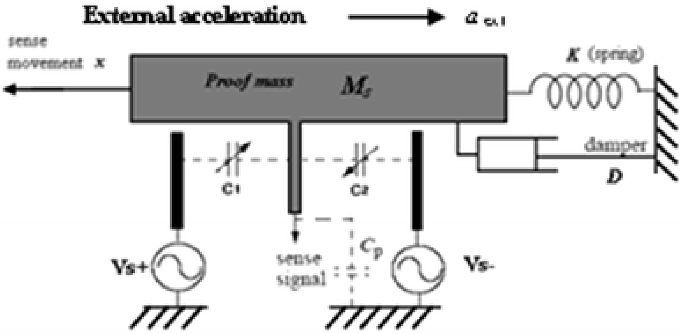

Figure 3.3.: Simplified diagram of MEMS-accelerometer

Piezoelectric The proof mass directly exerts its inertia onto a piezoelectric element, either by inducing an electric potential or by changing its resistance Howe und Cutkosky (1989).

4. Experiments

To detect the location of knocks on a table, a table was fitted with two Inverse MPU5600 senors, which are connected over an I2C bus with a Raspberry Pi Model 3 B+.

4.1. Set-up

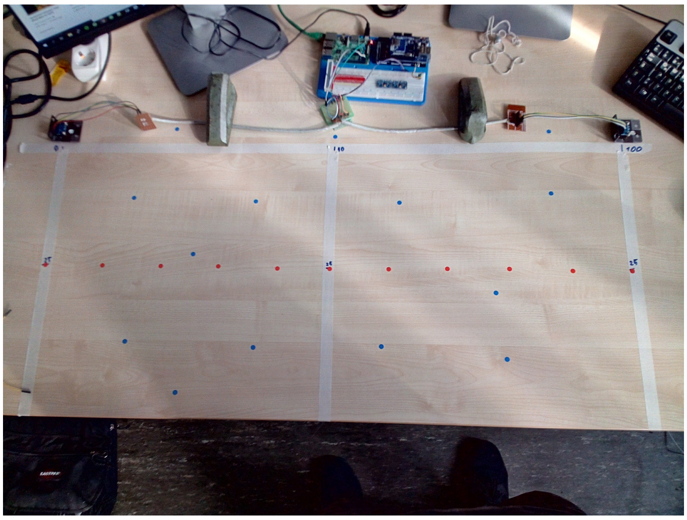

Figure 4.1.: Experimental set-up of the sensors on the table

The sensors are installed to a metal plate. The mass of the plate acts as a seismic mass, preventing the sensors from being influenced by minor disturbances. The metal plates are fitted with three feet that are kept from moving by adhesive tape. A three-legged design provides greater stability over a four-legged design.

The sensors are connected to the Raspberry Pi using Cat5e cables, reducing electromagnetic disturbances on the bus.

Due to the stiffness of the cables they were prevented from moving by weighting them down.

Sensor Placement The speed of sound in wooden panels ranges between $2700\frac{m}{s}$ in certain types of hardboard up to $3500\frac{m}{s}$ in massive oak panels. Typical parameters for hardboard are $c = 3000\frac{m}{s}$ for the speed of sound, a density of $\rho = 600\frac{kg}{m^3}$ and a modulus of elasticity of $E = 5.4 \cdot 10^9 \frac{kg}{ms^2}$ Lothar Cremer

The sample rate can be used to calculate the minimum distance between sensors and detect a propagation delay between the sensors.

$$\begin{aligned} \mathrm{f}_{Sensor} &= \text{sample rate} \\ c_L &= \text{Speed of sound} \\ d &= \text{distance between sensors} \quad = \frac{c_L}{\mathrm{f}_{Sensor}} \end{aligned}$$

with

$$\begin{aligned} c &= 3000\frac{m}{s} \\ \mathrm{f} &= 1500Hz \\ d &= \frac{c}{\mathrm{f}} = \frac{3000\frac{m}{s}}{1500\frac{1}{s}} \\ &= 2m \end{aligned}$$

To detect a delay of 1 sample, the sensors need to be placed at least $2m$ apart from each other. As this exceeds the size of an average tabletop, this approach will not be followed further.

For a more practical experiment the sensors were placed $1m$ apart, $0.5m$ from the long side of the table.

4.1.1. Tabletop

The tabletop has the dimensions $2000mm \cdot 900mm \cdot 250mm$ and seems to be made of multiple layers of hardwood. It is resting on two bars and is additionally hindered from oscillating by two further fixation points in the middle.

4.1.2. Sensors

Two Invensense MPU6500 were used. They are fitted with a 3-axis accelerometer and a 3-axis gyroscope, a typical low-cost motion processing unit for mobile devices. It comes with several features such as preprocessing, wake-on-motion and a low-power mode, which are not used here.

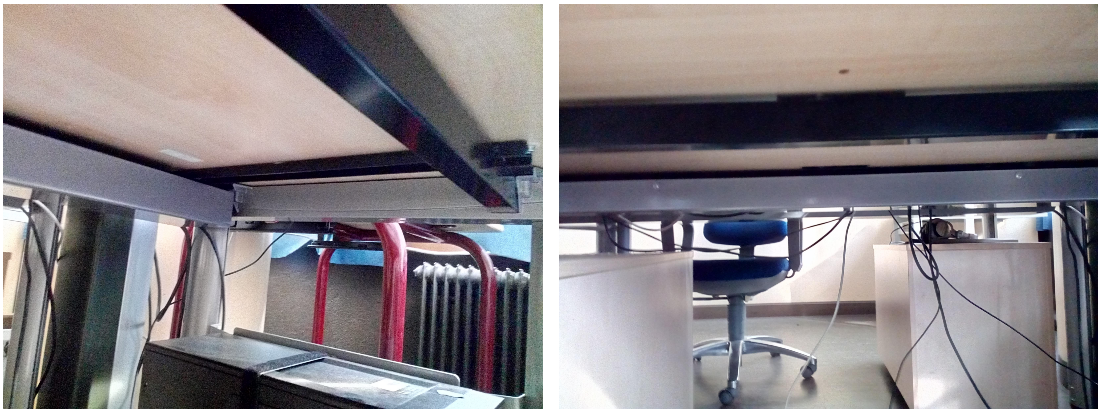

Figure 4.2.: Tabletop resting on the frame

The configuration and data registers are accessible via the I^{2}C interface. Each value consists of 16 bits split over two adjacent registers and is stored in two's complement format. InvenSense (2015)

Digital Motion Processor

The Digital Motion Processor implements the features mentioned above. If left to default settings it also applies a digital low-pass filter (DLPF) to the signal, smoothing the signal at the cost of a lower sample rate.

Gyroscope

Depending on the configuration the effective measuring range of the gyroscope is ±250°/s, ±500°/s, ±1000°/s and ±2000°/s. Its native sample rate is $8kHz$, but even when smoothing the signal with the DLPF it still outputs 3600 samples per second. Start-up time from sleep mode is 35 ms.

Accelerometer

The accelerometer can be configured to cover ±2g, ±4g, ±8g and ±16g measuring ranges. Default output sample rate is $500Hz$ with DLPF enabled and $4000Hz$ when the DLPF is bypassed. Start-up time is 30 ms.

Temperature

The temperature sensor can be used to calibrate the sensor in settings with a temperature much higher (or lower) than room temperature, as those conditions might influence the accuracy of the MEMS.

I^{2}C

The sensors primary interface is I^{2}C. By connecting a pin on the breakout board to V_0 its address can be incremented by 1, allowing two sensors of the same type to share one bus. The fastest supported transfer mode is fast-mode, allowing $400 kbit/s$.

Other Features

The following features were considered to be used, but ultimately discarded.

I^{2}C Master Mode The chip can act as an I^{2}C master, buffering values read from slave devices in its own registers. As the bus limits the output sample rate (4.2.1), sending more data over the same bus would reduce the effective sample rate even further.

FIFO An internal first in first out (FIFO) queue allows buffering up to 512 bytes. If we wanted to save all axes we could buffer 85 datasets in the FIFO. This was considered too small a sample size to do meaningful analysis.

4.1.3. Raspberry Pi

A Raspberry Pi 3 Model B+ is used to collect and analyse the data. The sensors are connected to the Raspberry via the I^{2}C-bus. Power is also delivered by the Raspberry.
Raspberry Pi Foundation (2014)

Hardware

CPU Broadcom BCM2836B0 Arm Cortex-A53 64-bit Soc @1.4 GHz

RAM 1GB LPDDR2 SDRAM

GPIO The Raspberry Pi can be expanded using the 40 exposed GPIO pins. Unfortunately only one I^{2}C-Bus is exposed, so both sensors will have to share a bus.

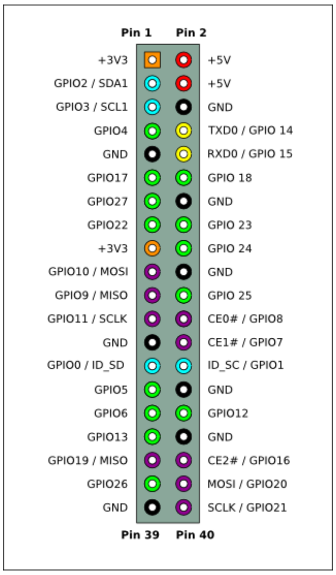

Figure 4.3.: Raspberry Pi Model 3 B+ GPIO header
Harvey

Software

Operating system The Raspberry Pi runs Raspbian, a version of the popular Debian GNU/Linux distribution.

Python All software is written in the Python programming language (version 3.5) The code is interpreted by the CPython reference implementation by www.python.org (Van Rossum und Drake Jr, 1995, cf)

Modules To capture the output of the sensors and interpreting the signals multiple third party python modules were used. Jones u. a. (2001–)

- matplotlib
- numpy
- scipy
- smbus
- scikit-learn

4.1.4. Data collection

Each dataset consists of 4000 samples. A python program running on the Raspberry Pi polls the sensors sequentially. Due to the limited data rate on the shared bus the transmission rate of the readings to the control unit is below the actual sample of $4kHz$ and therefore polling both sensors as fast as possible is the best option to get the maximum sample rate (4.1). Using the data-ready interrupt provided by the MPU-6500 was not an option because the execution time of the ISR would surpass the rate of incoming interrupts and the sensors kept hogging the bus and blocking each other.

After each capture period the data is stored in CSV-file. Due to the limited processing power of the Raspberry Pi the data was processed externally.

4.2. Sensor Configuration

To determine the optimal sensor configuration, several experiments with varying settings where conducted. This section describes the relevant insights gained through those experiments.

4.2.1. Sample Rate

A dataset consists of values for 6 axes plus temperature with a length of 2 bytes each, so total length of 112 bits. Because the two sensors share a bus the calculated sample rate must be divided by 2.

$$\frac{400000\frac{bit}{s}}{112bit} = 3571.4\frac{1}{s} \tag{4.1}$$

$$\frac{3571.4\frac{1}{s}}{2} = 1785.7Hz \tag{4.2}$$

Additionally the overhead of addressing the device and register (one byte each) and acknowledgements cost additional time.

Multi-Byte Reading

The Linux I^{2}C driver smbus supports the I^{2}C bus due to similarities in the SMBus and I^{2}C bus protocol. In contrast to byte wise reading, where every register has to be addressed individually, the master requests a number of bytes following the start register, therefore reducing control overhead and leaving more cycles for data transfer for the bus.

The temperature sensor's data register is unfortunately placed between the two axis sets of the inertia sensors. To skip those 16 bits would cost more time on the bus for sending control commands.

4.2.2. DLPF

Gyroscope

The DLPF can be configured to allow an output sample rate of $3600Hz$ for the gyroscope data, well above our $1.7kHz$ (InvenSense, 2015, 4.5)

Accelerometer

The maximal output sample rate of the accelerometer is $460Hz$ when using the DLPF, so the best solution is to bypass the DLPF for the accelerometer data path allowing a maximal output sample rate of $4000Hz$ (InvenSense, 2015, 4.8)

4.2.3. Measuring Range

The measuring ranges for both inertial sensors can be configured individually.

Gyroscope

As very small readings on the gyroscope axes are expected, using the minimal range of ±250°/s seems warranted. The least significant byte (LSB) corresponds to $\frac{1}{131^{\circ}/s}$ (InvenSense, 2015, Table 1)

Accelerometer

The accelerometers Z-axis measuring range is limited naturally by the $1g$ offset of earth's gravity, so when using the smallest measuring range $\pm 2g$ accelerations greater $1g$ might exceed the scale. The impulse created by knocking on the table presumably creates just a fraction of that. The LSB corresponds to $\frac{1}{16{,}384g}$ (InvenSense, 2015, Table 2)

4.3. Measurements

According to the Nyquist sampling theorem the sampling frequency should be greater or equal the highest frequency of the signal. The sensors output rate is far below the expected frequencies of an acoustic signal. (Mishali und Eldar, 2010, p 4)

Reducing the Number of Bytes to be Transferred

To increase the effective sample rate we could read fewer axes. The accelerometers Z axis and the two horizontal gyroscope axes were chosen to be read as they seem to be the most relevant.

4.3.1. Sensing Area

After finding the best configuration, sample data was collected.

25 points were chosen within the $50cm \cdot 100cm$ area between the sensors. A coordinate system was defined so the left sensor A is located at the origin $(0, 0)$, the right sensor B is located at $(100, 0)$ 10 of those points are located parallel to the x-axis at y = 25. In case location in two dimensions proves to be too difficult, those points can be used to try to at least determine the location along the x-axis. (See 4.1)

For each of the points P 20 measurements were taken, 10 with the full dataset, 10 with the reduced data set, each consisting of 4000 samples.

The tap events were given with one finger. This leads to varying intensity for the impulses.

5. Analysis

To detect the position of a knock on a table, the following techniques were considered.

5.1. Preprocessing

To smooth the signal recorded by the sensors and to analyse the data some preprocessing was required.

5.1.1. Savitzky-Golay Filter

The Savitzky-Golay filter was described first in Savitzky und Golay (1964). It's a commonly used filter in digital signal processing to increase the signal to noise ratio (SNR). Compared to a simple moving average filter the advantages the peaks get less deformed. Press u. a. (1992)

$$Y_j = \sum_{i=-\frac{m-1}{2}}^{\frac{m-1}{2}} C_i \, y_{j+i}, \frac{m-1}{2} \leq j \leq n - \frac{m-1}{2} \tag{5.1}$$

This project uses the `scipy.signal.savgol_filter` implementation Jones u. a. (2001–). All further operations were applied to the smoothed signal.

Mean

The mean is required to determine the offset for standardizing the signal. this project uses the `numpy.mean` implementation.

5.1.2. Standardization

The sensors do not read mean zero even when resting. While the AcZ axis is expected to measure gravity at around $9.81\frac{m}{s^2}$, all other axes also have some offset. To standardize the signal the mean is subtracted from every reading.

5.1.3. Signal Detection

To detect the impact event the maximal amplitude of the accelerometer's Z axis is assumed to be the start of the signal.

5.1.4. Planar Components

As the position of the origin of the signal is known to be in the plane of the x and y components of the sensors, we can calculate a vector $v_{\vec{x,y}}$ with the scalar length r (5.3) and the angle θ (5.4).
The vector is used when trying to read the bearing of the origin directly, as features for clustering only the 20 values with the greatest r are used.

$$v_{\vec{x,y}} = x_{\vec{x,0}} + y_{\vec{0,y}} \tag{5.2}$$

$$r = ||\vec{v}|| = \sqrt{||\vec{x}||^2 + ||\vec{y}||^2} \tag{5.3}$$

$$\theta = \arccos \frac{||\vec{x}||}{r} \tag{5.4}$$

5.2. Feature Extraction

The selection of features is based on the in (Braun u. a., 2015, p 3) described feature set. The microphone used in their paper provides a much higher sample rate, which can hopefully be compensated for in this work by looking at 5 more axes.

5.2.1. Variance

The variance for each axis is calculated using the `numpy.var` implementation.

5.2.2. Standard Deviation

This project uses the `numpy.std` implementation.

5.2.3. Frequency Components

Fast Fourier Transform

To divide a signal into it's frequency components the fast Fourier transform algorithm is used. The algorithm divides a continuous signal into its frequency components.

$$f_m = \sum_{k=0}^{2n-1} x_k \, e^{-\frac{2\pi i}{2n} mk} \qquad m = 0, \dots, 2n-1 \tag{5.5}$$

Cochran u. a. (1967)

```
# n_samples, samplerate, values
freqs  = numpy.fft.fftfreq(n_samples, 1 / samplerate)
freqs  = numpy.fft.fftshift(freqs)

result = numpy.fft.fft(values)
result = numpy.abs(result))
result = numpy.fft.fftshift(result)
```

Compared with Braun u. a. (2015) the lower sample rate translates into a lower Nyquist-frequency up to about $400Hz$.

5.3. Bearing

To determine the real direction of the forces in the plane of the table, the X and Y components of the signal must be treated as a vector. Transferred to polar coordinates the angle θ at the largest elongations of $\vec{v_{x,y}}$ might be correlated to the x, y location of the source.

5.4. Machine Learning

There are two types of machine learning algorithms: Supervised and unsupervised algorithms. Supervised algorithms are used for classification and regression tasks, based on whether the data is discrete (classification) or continuous (regression). Supervised algorithms require that the right classifications are known and that the data set is labelled with those classifications. The data is usually split into a training set and a test set. The algorithm then learns a model for classification or regression based on the training data. The accuracy and precision of the found model is then evaluated on the test set.

In unsupervised learning, the data set is not labelled. Unsupervised algorithms split the data into groups based on found similarities. What constitutes these similarities depends on the algorithm that is used. In clustering, some algorithms require that the number of groups is known beforehand, while others find the number of clusters themselves.

5.4.1. Unsupervised

To find relations between a big number of features clustering methods are used.

At first it is evaluated if the measurements taken for the 25 individual locations will be clustered into the sets they belong to, in which case the chosen feature set contains enough information to create distinct clusters. The features are transformed into a vector which denotes a point in a multi-dimensional space, where the number of dimensions is equal to the number of features. To find groups or clusters of data different algorithms can be used.

K-means Clustering

K-means is a very simple algorithm to find clusters. It was popularized by MacQueen (1967). there are different implementations, but Lloyd's algorithm is the most common implementation. For each assumed cluster a centre point is assumed within the data range. The initial position of those centroids should be chosen so their distance from each other is maximized. Each data point is then assigned to its nearest centre point (5.6). The centroid is then moved to the centre of the cluster (5.7). these steps are repeated until the assignments of the data points no longer change. Steinbach u. a. (2000)

$$S_i^{(t)} = \{x_p : ||x_p - m_i^{(t)}||^2 \leq x_p - m_j^{(t)}||^2 \qquad \forall j, 1 \leq j \leq k\} \tag{5.6}$$

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum x_j \in S_i^{(t)} x_j \tag{5.7}$$

Clustering was done for the X and Y component of the location separately, using the `sklearn.cluster.KMeans` implementation.

Agglomerative Hierarchical Clustering

Hierarchical cluster analysis provides a more flexible approach than K-means clustering. Initially every feature vector is considered to form its own cluster. By agglomerating clusters iteratively until the requested number of clusters is reached. While K-means clustering only considers the euclidean distance of each data point to the cluster centroids, agglomerative hierarchical clustering can use different measures as a distance and linkage function between two clusters. As clusters aren't defined by a centroid, but the set of data points they contain, linkage measurement can be calculated using the minimal distance between any two points (Single Linkage), the largest distance between two points (Complete Linkage) or any weighted combination of the distances of multiple point pairs (Ward). Euclidean and Manhattan distance were used as distance measures with multiple linkage strategies each. The `sklearn.cluster.AgglomerativeClustering` implementation was used. Steinbach u. a. (2000)

5.4.2. Supervised

If matching the measurements to the coordinates of the acoustic source succeeds, supervised learning approaches can be used to evaluate whether the feature vector can be reduced to the continuous two dimensional values of the location.

6. Results

Analysis of the data was first done separately for the measurements over all axes (Normal) and the reduced set of axes AcZ, GyX, GyY (Fast), then the datasets were combined (All).

6.1. Triangulation

Triangulating the event source was unsuccessful. Adding x and y components of the data as vectors and using the calculated angles α and β (3.1) to find the bearing of the tapping did not yield the desired results. Linear interpolation of the vectors with the greatest length r was not strongly correlated to the location (B.1). The intersection points fluctuate around the (50, 25) position, but are too weakly correlated to the location to determine a definitive trend
As B.1 shows with the two points $P_{0,25}$ and $P_{100,25}$, chosen because they are aligned with the sensors y axes, the angles relative to each other hardly change. The noticeably greater spread of the left sensor can be explained by the left sensor being fixed to the table differently than the right one. The three-legged support of the sensors might reduce the sensitivity of the sensor by observing the deflection not in a single point but as a function of the thee contact points to the table. The centre of the sensing area also is roughly in the middle between the two bars supporting the tabletop, increasing its ability to oscillate more freely.

6.2. Machine Learning

The different machine learning strategies also yielded no result. While the problems mentioned above probably also apply here, the number of samples is also rather small compared to the complexity of the data. Neither K-Means clustering nor hierarchical clustering succeeded in classifying measurements taken for the same point into the same cluster. The found clusters seem to be totally random, but might represent some hidden variable. As the tap events were not standardized and their intensity was not recorded this might be the hidden variable the clusters are found for.

6.2.1. K-Means Clustering

As shown in A.1 the algorithm can't find many similarities between clusters for the full dataset. Fo the reduced axes set results seem more promising (A.2): $P_{0,88}$ can be predicted for 10 of the 10 available measurements when using the other nine as training data. $P_{0,50}$ and $P_{0,21}$ are close as well with 8/10 and 7/10 correct predictions respectively. Interestingly the 3 false predictions for $P_{0,21}$ are clustered with $P_{0,50}$. This suggests that the gyroscope data might be useful to detect the location of tap events on the y-axis.

6.2.2. Agglomerative Hierarchical Clustering

Agglomerative hierarchical clustering yielded vastly different results depending on the distance function used (A.4). In contrast to K-Means clustering, the measurements are spread unevenly among the clusters, overpopulating a few clusters while leaving many containing only one or zero measurements. This is a strong indicator that the set of features chosen was not distinct enough. A better selection or normalisation of the chosen features might yield better results.

6.2.3. Supervised learning

The regression analysis was not conducted, as the unsupervised clustering strategies proved unsuccessful and the sample set was considered to be too small to result in a meaningful model.

7. Conclusion

This thesis is unable to give a conclusive statement whether consumer-grade inertia sensors can be used to detect and localize input on tabletops.

The material and construction of the table, as well as the objects on top of it, have a larger influence on its acoustic properties than anticipated. This problem might be mitigated if the sensors provided a higher sample rate to distinguish between the real signal and its reflections.

Set-up The influence of the locations of the suspension points of the table top was underestimated. Using an empty table without multiple heavy objects on top of it would decrease random scattering due to refraction where the thing on top exerts pressure onto the plate influencing its density.

Signal Generation The unequal intensity of the tap event should not matter in the end, but using a tap with predefined strength would make comparison of the readings much easier.
This could be combined with an interrupt to the Raspberry to start capturing immediately, as opposed to the system used here where the system informs the user to prepare giving an input during the sample period.

Hardware While the three legged design of the mounting plates may prevent the sensors from wobbling along the axis of two feet it also greatly reduces the spacial resolution of the sensor. By resting on three legs approximately 3 cm from each other the sensor can only detect common wave fronts as belonging together if they arrive perpendicular to the line between the legs. Overall the sample rate and sensitivity reached by low-cost inertia sensors is far below the required sample rate and the reflections of the vibrations within the table are far too random to accurately detect the source of a tap. While their frequency ranges might be just enough the limited throughput of the bus is hindering using them to their full potential. The effects of under sampling are obvious even in the raw data and distort the data (4.3). Using a dedicated I^{2}C bus or I^{2}C high-speed-mode for each sensor would be more preferable. As described in 6.2.1 there are significant results under special circumstances.

Real Time Data Collection Because the data capture was run as a user process on the system and simply relied on polling as fast as possible the script could be interrupted by the operating systems scheduler, leading to uneven spacing between the samples. This makes the Savitzky-Golay filter (5.1.1) and the frequency analysis (5.2.3) unreliable. Using an interrupt, either triggered by a timer or provided by the sensor to read data with constant sample distance would help a lot

Calibration Even if these problems can be mitigated, calibration still would be an issue to deal with, as the propagation of acoustic waves depends on the material of the table, location and form of the support points, and objects placed on the table, as their weight will change the tensions and forces within the tabletop and thereby its acoustic properties. This would require retraining the machine learning models for every change to the surface.

A. Tables

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)		1	1	3	2		1			1	3		2							3	1			1	1
(10, 25)		4		3			1			2	1		4							2	2		1		
(15, 15)					5		1		1		7		4												2
(16, 39)	3	1		6			1			2	1		2			1					1		2		
(20, 25)		2		2	1		4						4						1	2	1	3			
(21, 0)				3			1		6		6														4
(24, 24)		2		6	1					5	2		1							1	1				1
(24, 46)	3			3		1	5	1					2			3	1	1							
(30, 25)	1	1		6	1		2					1	2							3	1	2			
(36, 15)				8	4					2	2										3				1
(39, 39)	2			2		5	3						5			1	1				1				
(40, 25)				4	1		2			1	1					1					2	1	4		3
(50, 0)				2	2		7		3		4														2
(50, 25)				1	3		4			1			2								1		4		4
(59, 40)	4			1		1	4						4		1						2		3		
(60, 25)				6	5		1									1					2		3		2
(62, 15)				3	4						6		1								2		2		2
(70, 25)				6	4						3		1								2		4		
(78, 30)	2			5						1	5			1		1							5		
(79, 42)	1			5		1	4			1			2			5							1		
(80, 25)	1			5	2		1				3										2		6		
(88, 0)									12		7		1												
(88, 12)				3	4						6		3								3		1		
(90, 25)				1	2						7		1			1					4		4		
(100, 25)				1	5				1	1	6		1										3		2

Table A.1.: All - KMeans

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)	1				1						3							1	1			2			1
(10, 25)											2								2			2		1	3
(15, 15)	8																					1	1		
(16, 39)							1			4				1			1		2		1				
(20, 25)	1								1		3											1		1	3
(21, 0)						7																	3		
(24, 24)							1							2					5			1			1
(24, 46)		1	2	1						1						1	2			1	1				
(30, 25)	1						2	1		1	2			1					1			1			
(36, 15)	3					1	1			1				1					1			2			
(39, 39)		3	1							1				1	2		1			1					
(40, 25)	1						3												2		1	2		1	
(50, 0)	2																						8		
(50, 25)	3					1	3												1			1		1	
(59, 40)		1	1							4			1	1										2	
(60, 25)	4						2							2								1		1	
(62, 15)	5						2															2	1		
(70, 25)	3						1							1					1			1		3	
(78, 30)			1				4			1		1							2		1				
(79, 42)		1	2			1	1			1							1		1		2				
(80, 25)	2						4							1								1		2	
(88, 0)						10																			
(88, 12)	5					2								2								1			
(90, 25)	2						1							3								2		2	
(100, 25)	3					1	2							1					1			1		1	

Table A.2.: Fast - KMeans

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)		2		1	2							1					1					2		1	
(10, 25)					4		1					1					1							3	
(15, 15)		1		2						1		3										3			
(16, 39)		2							1			1					4	2							
(20, 25)					4	1		1	1															3	
(21, 0)				3				1		3				3											
(24, 24)		6		1	1																			2	
(24, 46)					1	3			1								2	1						2	
(30, 25)		2			1				2								1	1						2	1
(36, 15)		5		1				2									2								
(39, 39)		1			4	2			1									1						1	
(40, 25)				3				1	2			1					1							2	
(50, 0)	1	1		2				2					1									2	1		
(50, 25)				4				1	4									1							
(59, 40)		1			1				4						1			2		1					
(60, 25)				2				3	1										1					3	
(62, 15)				2				3														5			
(70, 25)		5			1			1				1										1		1	
(78, 30)		2						2				1										3		2	
(79, 42)		1															2	2						5	
(80, 25)								5	1			1				1						2			
(88, 0)	2									2				3							3				
(88, 12)								4			1	1					1					3			
(90, 25)		2	1		1			1									1					2			2
(100, 25)	1			2				2				2						1				2			

Table A.3.: Normal - KMeans

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)	2			13						3					1				1						
(10, 25)	7			11						2															
(15, 15)		1		19																					
(16, 39)	7			11				1														1			
(20, 25)	2			12						4			1			1									
(21, 0)		6		14																					
(24, 24)	7			13																					
(24, 46)			1	10	1			3									1	1		1		1	1		
(30, 25)	3			13				1		2											1				
(36, 15)	2			18																					
(39, 39)			2	10	1	1		2						1								1		1	1
(40, 25)	7			13																					
(50, 0)		3		17																					
(50, 25)	5			15																					
(59, 40)	5			11				2				1													1
(60, 25)	3			16					1																
(62, 15)	2			18																					
(70, 25)	5			15																					
(78, 30)	7			10			1	2																	
(79, 42)	3		1	11				3										1				1			
(80, 25)	6			13							1														
(88, 0)		12		8																					
(88, 12)	1			19																					
(90, 25)	4			15					1																
(100, 25)	4	1		15																					

Table A.4.: All - Agglomerative Euclidean Average

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)				3					3						1				1						2
(10, 25)				2					2																6
(15, 15)				10																					
(16, 39)	1			1			1			1															6
(20, 25)				2					4				1	1											2
(21, 0)				10																					
(24, 24)				3																					7
(24, 46)	1	1	1				3			1							1	1					1		
(30, 25)				3			1		2												1				3
(36, 15)				8																					2
(39, 39)		1	2			1	2			1	1									1				1	
(40, 25)	1			3																					6
(50, 0)				10																					
(50, 25)				5																					5
(59, 40)				1			2	1				1													5
(60, 25)				7																					3
(62, 15)				8																					2
(70, 25)				5																					5
(78, 30)	1						2									1									6
(79, 42)	2		1				3			1												1			2
(80, 25)				4																					6
(88, 0)				8	2																				
(88, 12)				10																					
(90, 25)				6																					4
(100, 25)				6																					4

Table A.5.: Fast - Agglomerative Euclidean Average

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)								3		7															
(10, 25)								4		5									1						
(15, 15)						2				7				1											
(16, 39)								2		7	1														
(20, 25)							1	4		4	1														
(21, 0)	1		3			3				1															2
(24, 24)						1		1		8															
(24, 46)							3	2		4	1														
(30, 25)								2		5	2							1							
(36, 15)						1				8	1														
(39, 39)							2	5		2	1														
(40, 25)						3				5	2														
(50, 0)					1	2				5											1			1	
(50, 25)						4		1		1	4														
(59, 40)								3		1	4		1				1								
(60, 25)						2				6	1									1					
(62, 15)						2				8															
(70, 25)								1		9															
(78, 30)										10															
(79, 42)								2		8															
(80, 25)										5	4	1													
(88, 0)	2	2		2	2				1													1			
(88, 12)										9					1										
(90, 25)								1		8						1									
(100, 25)						2		1		6													1		

Table A.6.: Normal - Agglomerative Euclidean Average

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)	11		2							3		1			1			1							1
(10, 25)	9		2	2						2								4							1
(15, 15)	15		3			1																		1	
(16, 39)	9	1	1	6				1																1	1
(20, 25)	8		2				1			4								2					1	1	1
(21, 0)	11					6																		3	
(24, 24)	11		2	4														1							2
(24, 46)	6	2					3	2	1		1		1	1					1			1		1	
(30, 25)	9		2	1				1		2											1			2	2
(36, 15)	13		3	1																				1	2
(39, 39)	7	1			1		2	2	1		2									2		1		1	
(40, 25)	9		2	6																				2	1
(50, 0)	11					3																		6	
(50, 25)	10		1	4																				4	1
(59, 40)	5		2	4	1			3									1							4	
(60, 25)	12		3	3										1										1	
(62, 15)	16		2	2																					
(70, 25)	13		2	4						1															
(78, 30)	10			6				1						1		1									1
(79, 42)	10	2		2			1	1			1			2											1
(80, 25)	7		2	6				1																4	
(88, 0)	8					12																			
(88, 12)	16		3	1																					
(90, 25)	11		4	3						1				1											
(100, 25)	12		2	3		1				2															

Table A.7.: All - Agglomerative Euclidean Complete

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)		1		3				2		1					1								1		1
(10, 25)	2			2				2		1															3
(15, 15)		6						3												1					
(16, 39)	6		1				1			1	1														
(20, 25)				4				2				1													3
(21, 0)		7																		3					
(24, 24)	4	1						1		2	1														1
(24, 46)			2			1	2		1				1						1			1		1	
(30, 25)	1	1		2			1	1			1										1				2
(36, 15)	1	4						2		2	1														
(39, 39)			1		1	1	2		1									2				2			
(40, 25)	6	1						2																	1
(50, 0)		2																		8					
(50, 25)	4	4						1		1															
(59, 40)	4				1		3				1						1								
(60, 25)	3	4						1			2														
(62, 15)	2	5						2												1					
(70, 25)	4	3		1				1			1														
(78, 30)	6		1				1			1						1									
(79, 42)	2	1	3				1			1												1		1	
(80, 25)	6	2						1			1														
(88, 0)		8												2											
(88, 12)		7						1			2														
(90, 25)	3	2		1				1			3														
(100, 25)	3	3		2				2																	

Table A.8.: Fast - Agglomerative Euclidean Complete

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)						1	2	1			3											2			1
(10, 25)						4	3	1															1		1
(15, 15)											7	2					1								
(16, 39)			1			2	6				1														
(20, 25)			1			2	2							1								2			2
(21, 0)	4				2		1					3													
(24, 24)							3				4	1										1			1
(24, 46)			2			1	3							3								1			
(30, 25)			2			2	3				2													1	
(36, 15)			1				4				4	1													
(39, 39)						2	2															3			3
(40, 25)			2				2	1				3													2
(50, 0)				1				2			3	2			1				1						
(50, 25)			4				1					4										1			
(59, 40)			2			3					1									1	1				2
(60, 25)			1				6					2	1												
(62, 15)							1	2			5	2													
(70, 25)						1	4				5														
(78, 30)							4				6														
(79, 42)						2	6				1														1
(80, 25)			4				2	1			2					1									
(88, 0)	2	3		2	1					2															
(88, 12)							1	6	1		2														
(90, 25)							1	4			3							1				1			
(100, 25)				1				3			3	2										1			

Table A.9.: Normal - Agglomerative Euclidean Complete

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)		1	2		1	1		5		1	2		2	3	1								1		
(10, 25)		1			2	2		4		4	3		2	2											
(15, 15)			8	1				7						1		2								1	
(16, 39)	1	2			1	4		7						2										1	2
(20, 25)			1		1	1	1			2	4		4	4						1				1	
(21, 0)				6				6			4					3								1	
(24, 24)		1	1		2	5		5		1	3			1		1									
(24, 46)	3						3				4	1		2				1	1			1		1	3
(30, 25)	1	2	1		2	2		1			5		1	2			1							2	
(36, 15)		1	3		3	2		4			6					1									
(39, 39)	2						2				2	5		5				1						1	2
(40, 25)		3	2		2	2		1			4					3								2	1
(50, 0)			2	3				3			4					2								6	
(50, 25)		4	3		1	1					1			2		4								4	
(59, 40)	1	6			2						1	1		4							1			4	
(60, 25)	1	3	4		3						6					2								1	
(62, 15)		2	5		2			5			4					2									
(70, 25)		4	4		2			5			4			1											
(78, 30)	2	5				1		5	1		5														1
(79, 42)	3	1				1	1	1			6	1		2										1	3
(80, 25)	1	6	2		2			2			6													1	
(88, 0)				12				7			1														
(88, 12)		1	4		3			5			6			1											
(90, 25)	1	4	2		4			6			2			1											
(100, 25)		4	4	1		1		4			3			1		2									

Table A.10.: All - Agglomerative Euclidean Ward

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)					2			1					1	1					1			1	1	1	1
(10, 25)					2			3						2								1			2
(15, 15)		1										1	5											3	
(16, 39)	2			1		1			1												1				4
(20, 25)					4			2			1			1		1								1	
(21, 0)		7										3													
(24, 24)	1							1					1	1							1				5
(24, 46)			1	2		3			1	1							1	1							
(30, 25)	1				1	1					2		1	1						1	1	1			
(36, 15)	1	1											3	2							1				2
(39, 39)			1	2		2				5															
(40, 25)	3								1		1		2	2											1
(50, 0)												8	2												
(50, 25)	4	1											3	1											1
(59, 40)	6					1				1					1						1				
(60, 25)	3												4	1							2				
(62, 15)	2											1	5	2											
(70, 25)	3												4	1							1	1			
(78, 30)	5					2	1		1																1
(79, 42)	1			1		3			2	1		1													1
(80, 25)	4												2	1							1	2			
(88, 0)		10																							
(88, 12)		3											4	1							2				
(90, 25)	2												2	1							3	2			
(100, 25)	4	1											4												1

Table A.11.: Fast - Agglomerative Euclidean Ward

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)	2	2						1	1							1	3								
(10, 25)				2		1		1	1							3	1						1		
(15, 15)	7													1					2						
(16, 39)		1	1	1													7								
(20, 25)		3	1	1		2			2				1												
(21, 0)							1							3					3						3
(24, 24)		1					3		1								4		1						
(24, 46)		2	1			1		2	1				3												
(30, 25)			2	2		1	2	3																	
(36, 15)	1					2	3	2									1		1						
(39, 39)		3	1	2		2							2												
(40, 25)			2			1		1	2							1			3						
(50, 0)	2				1	1	2												2	1				1	
(50, 25)		1	4			1													4						
(59, 40)		1	4	2			1								1			1							
(60, 25)			1			6													2			1			
(62, 15)	3					2	3												2						
(70, 25)	1			1		3	3										2								
(78, 30)	3					4	3																		
(79, 42)			1	2		3		2	1								1								
(80, 25)	2		1			5	1					1													
(88, 0)					2					3				2											3
(88, 12)	4					2	2	1			1														
(90, 25)	2						3									3	1				1				
(100, 25)	3	1			1		3												2						

Table A.12.: Normal - Agglomerative Euclidean Ward

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)															1			18			1				
(10, 25)					1													19							
(15, 15)														1				19							
(16, 39)																		20							
(20, 25)																		19	1						
(21, 0)	3		2							1								14							
(24, 24)																		20							
(24, 46)				1			1						1				1	16							
(30, 25)																		19					1		
(36, 15)																		20							
(39, 39)		1		4			1											14							
(40, 25)																		20							
(50, 0)												1						17		1					1
(50, 25)																		20							
(59, 40)		1							1									17						1	
(60, 25)																		19				1			
(62, 15)																		20							
(70, 25)																		20							
(78, 30)																1		19							
(79, 42)				1														19							
(80, 25)																		19				1			
(88, 0)	2		1			2		2		2	1							8							2
(88, 12)					1													19							
(90, 25)																		19				1			
(100, 25)																		19							1

Table A.13.: All - Agglomerative Manhattan Average

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)	8														1				1						
(10, 25)	10																								
(15, 15)	8								2																
(16, 39)	8			1																					1
(20, 25)	8																1							1	
(21, 0)									10																
(24, 24)	10																								
(24, 46)	3	1														1		1			1	1			2
(30, 25)	7				2																		1		
(36, 15)	9								1																
(39, 39)	2	1	1				1	1					1	1						1					1
(40, 25)	8			1	1																				
(50, 0)	2								8																
(50, 25)	9								1																
(59, 40)	8					1						1													
(60, 25)	10																								
(62, 15)	9								1																
(70, 25)	10																								
(78, 30)	7			2							1														
(79, 42)	2		1	6					1																
(80, 25)	10																								
(88, 0)									8	2															
(88, 12)	7								3																
(90, 25)	10																								
(100, 25)	9								1																

Table A.14.: Fast - Agglomerative Manhattan Average

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)			3		7																				
(10, 25)			5		4																	1			
(15, 15)					9																		1		
(16, 39)			2		8																				
(20, 25)			4		6																				
(21, 0)	1				4								1				1	1		1				1	
(24, 24)			1		9																				
(24, 46)			2		8																				
(30, 25)			3		7																				
(36, 15)					10																				
(39, 39)			5		5																				
(40, 25)			1		9																				
(50, 0)					7			1			1	1													
(50, 25)			1		9																				
(59, 40)			4		5														1						
(60, 25)					9	1																			
(62, 15)					10																				
(70, 25)			1		9																				
(78, 30)					10																				
(79, 42)			2		8																				
(80, 25)					9				1																
(88, 0)	1	2		1						1				1		1					1				2
(88, 12)					9		1																		
(90, 25)			3		6	1																			
(100, 25)			1		8										1										

Table A.15.: Normal - Agglomerative Manhattan Complete

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)		5								3	10				1	1									
(10, 25)		4								6	9										1				
(15, 15)		9									10						1								
(16, 39)										9	10									1					
(20, 25)		4								5	10							1							
(21, 0)	4										14														2
(24, 24)		3								7	10														
(24, 46)				1			1			4	10		1	1						2					
(30, 25)		4								5	10												1		
(36, 15)		6								3	11														
(39, 39)				3					2	3	10			1						1					
(40, 25)		3								7	10														
(50, 0)		2			1						15											1			1
(50, 25)		5								5	10														
(59, 40)		1							1	8	9	1													
(60, 25)		7				1				3	9														
(62, 15)		7								2	11														
(70, 25)		5								5	10														
(78, 30)										9	10								1						
(79, 42)				1						8	11														
(80, 25)		4				1				6	9														
(88, 0)	2	1	3		2			2			7													2	1
(88, 12)		10									9										1				
(90, 25)		6				1				4	9														
(100, 25)		4			1					5	10														

Table A.16.: All - Agglomerative Manhattan Complete

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)				1		1				4		1			1					1					1
(10, 25)				3						4										1					2
(15, 15)						9										1									
(16, 39)		1																		4					5
(20, 25)				5						4			1												
(21, 0)																10									
(24, 24)				1		2				1										2					4
(24, 46)	1	2	1		3														1	1			1		
(30, 25)						1				3										1		2		1	2
(36, 15)						6										1				2					1
(39, 39)	2	1	1		1		1				1						1			1					1
(40, 25)						2				1										2		1			4
(50, 0)						2										8									
(50, 25)						5														1					4
(59, 40)									1									1			1				7
(60, 25)						7																			3
(62, 15)						7										1									2
(70, 25)						5																			5
(78, 30)								1	2											2					5
(79, 42)							1		5							1				1					2
(80, 25)						4																			6
(88, 0)						1								2		7									
(88, 12)						10																			
(90, 25)						6																			4
(100, 25)						4										1									5

Table A.17.: Fast - Agglomerative Manhattan Complete

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(0, 25)	10																								
(10, 25)	9																			1					
(15, 15)	9																		1						
(16, 39)	10																								
(20, 25)	10																								
(21, 0)	4	1													1		1	1				1	1		
(24, 24)	10																								
(24, 46)	10																								
(30, 25)	10																								
(36, 15)	10																								
(39, 39)	10																								
(40, 25)	10																								
(50, 0)	7						1	1								1									
(50, 25)	10																								
(59, 40)	9												1												
(60, 25)	9					1																			
(62, 15)	10																								
(70, 25)	10																								
(78, 30)	10																								
(79, 42)	10																								
(80, 25)	9																							1	
(88, 0)		1	1		1				1	1	1			1							1				2
(88, 12)	9			1																					
(90, 25)	9					1																			
(100, 25)	9											1													

Table A.18.: Normal - Agglomerative Manhattan Average

B. Plots

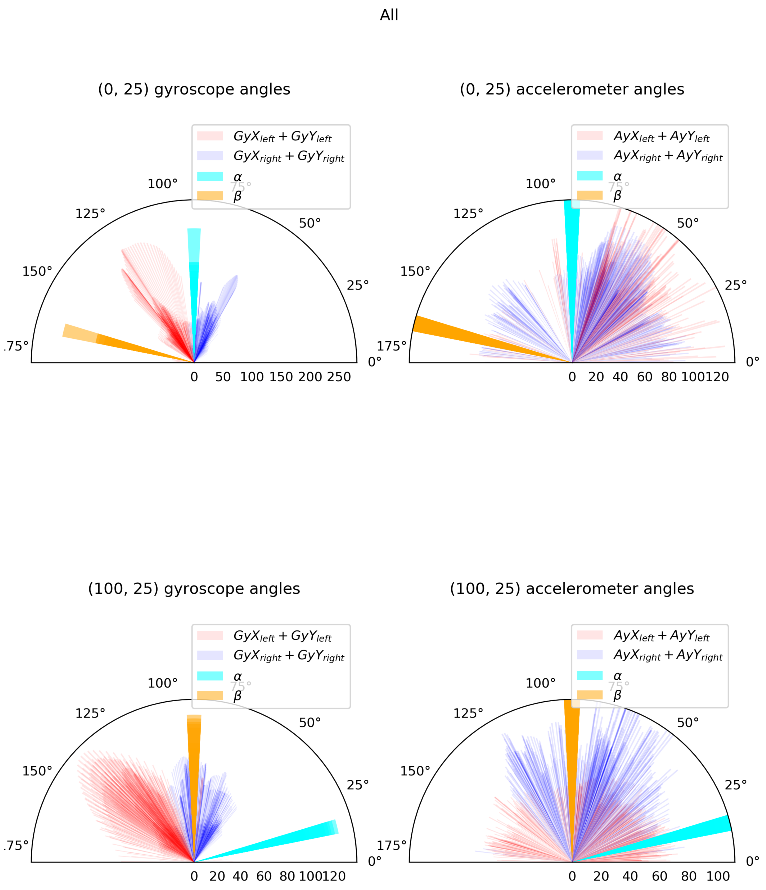

Figure B.1.: The angles of the 100 greatest scalar values for $\vec{x} + \vec{y}$

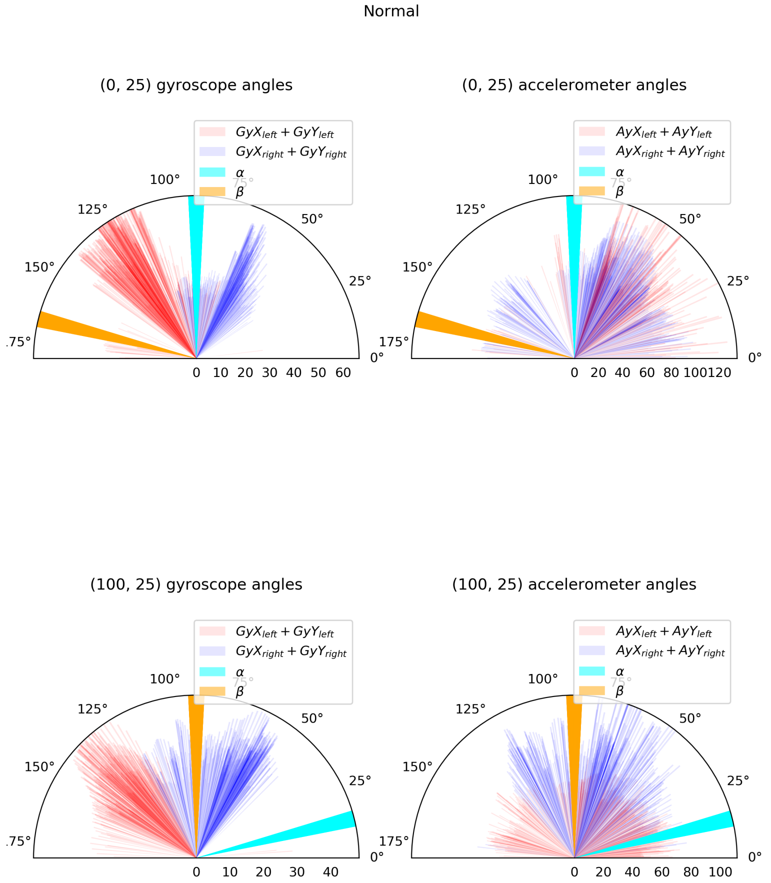

Figure B.2.: The angles of the 100 greatest scalar values for $\vec{x} + \vec{y}$

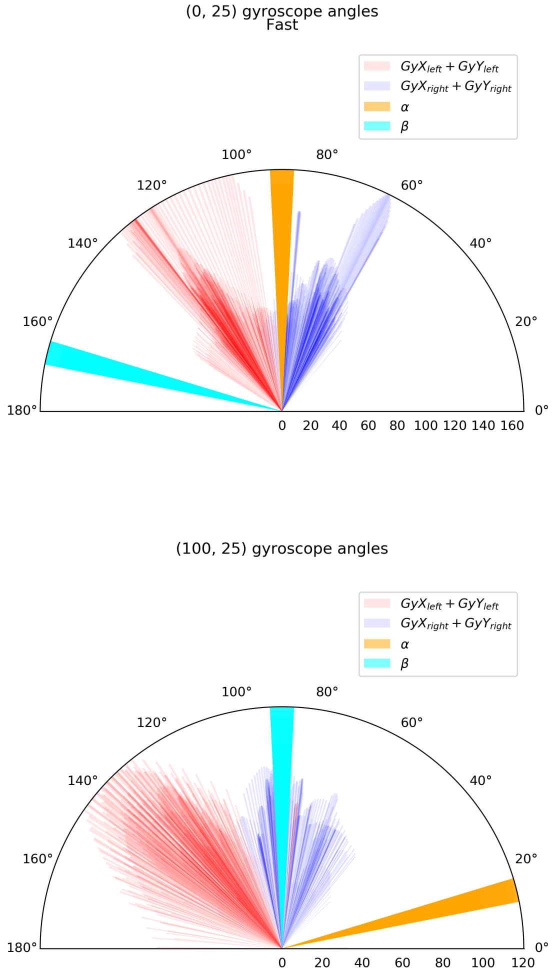

Figure B.3.: The angles of the 100 greatest scalar values for $\vec{x} + \vec{y}$

Bibliography

[Anstey 1977] Anstey, Nigel A.: *Seismic Interpretation: The Physical Aspects.* 1. Springer Netherlands, 1977. – URL http://gen.lib.rus.ec/book/index.php?md5=2EB4DAC397F1364DDA303CA72E4F9024. – ISBN 978-0-934634-18-2,978-94-015-3924-1

[Benali-Khoudja u. a. 2004] Benali-Khoudja, Mohamed ; Hafez, Moustapha ; Alexandre, Jean-Marc ; Kheddar, Abderrahmane: Tactile interfaces: a state-of-the-art survey. In: *Int. Symposium on Robotics* Bd. 31 Citeseer (Veranst.), 2004, S. 23–26

[Bernstein u. a. 2003] Bernstein, Jonathan u. a.: An overview of MEMS inertial sensing technology. In: *Sensors-The Journal of Applied Sensing Technology* 20 (2003), Nr. 2, S. 14–21. – URL https://www.sensorsmag.com/components/overview-mems-inertial-sensing-technology

[Braun u. a. 2015] Braun, Andreas ; Krepp, Stefan ; Kuijper, Arjan: Acoustic tracking of hand activities on surfaces. In: *Proceedings of the 2nd international Workshop on Sensor-based Activity Recognition and Interaction* ACM (Veranst.), 2015, S. 9

[Cochran u. a. 1967] Cochran, William T. ; Cooley, James W. ; Favin, David L. ; Helms, Howard D. ; Kaenel, Reginald A. ; Lang, William W. ; Maling, George C. ; Nelson, David E. ; Rader, Charles M. ; Welch, Peter D.: What is the fast Fourier transform? In: *Proceedings of the IEEE* 55 (1967), Nr. 10, S. 1664–1674

[Colegrove 2010] Colegrove, Jennifer: The state of the touch-screen market in 2010. In: *Information Display* 26 (2010), Nr. 3, S. 22–24

[Demtröder 1998] Demtröder, Wolfgang: *Experimentalphysik.* Bd. 1. Springer, 1998

[Deutsch u. a. 2013] Deutsch, Volker ; Platte, Michael ; Vogt, Manfred: *Ultraschallprüfung: Grundlagen und industrielle Anwendungen.* Springer-Verlag, 2013

[Diamant und Lampe 2013] Diamant, Roee ; Lampe, Lutz: Underwater localization with time-synchronization and propagation speed uncertainties. In: *IEEE Transactions on Mobile Computing* 12 (2013), Nr. 7, S. 1257–1269

[Dohse u. a. 2008] Dohse, K. C. ; Dohse, T. ; Still, J. D. ; Parkhurst, D. J.: Enhancing Multi-user Interaction with Multi-touch Tabletop Displays Using Hand Tracking. In: *First International Conference on Advances in Computer-Human Interaction*, Feb 2008, S. 297–302

[Harvey] Harvey, Ian: *Pi-GPIO-header*. – URL https://elinux.org/File:Pi-GPIO-header.png

[Howe und Cutkosky 1989] Howe, Robert D. ; Cutkosky, Mark R.: Sensing skin acceleration for slip and texture perception. In: *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on* IEEE (Veranst.), 1989, S. 145–150

[InvenSense 2015] InvenSense: *MPU-6500 Product Specification*. : , 2 2015. – URL https://www.invensense.com/wp-content/uploads/2015/02/MPU-6500-Datasheet2.pdf. – Revision 1.1

[InvenSense 2015] InvenSense: *MPU-6500 Register Map*. : , 2 2015. – URL https://www.invensense.com/wp-content/uploads/2015/02/MPU-6500-Register-Map2.pdf. – Revision 2.1

[Jones u. a. 2001–] Jones, Eric ; Oliphant, Travis ; Peterson, Pearu u. a.: *SciPy: Open source scientific tools for Python*. 2001–. – URL http://www.scipy.org/. – [Online; accessed 2018-06-22]

[Letessier und Bérard 2004] Letessier, Julien ; Bérard, François: Visual tracking of bare fingers for interactive surfaces. In: *Proceedings of the 17th annual ACM symposium on User interface software and technology* ACM (Veranst.), 2004, S. 119–122

[Lothar Cremer] Lothar Cremer, Manfred Heckl (.: *Körperschall: Physikalische Grundlagen und technische Anwendungen*

[MacQueen 1967] MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif. : University of California Press, 1967, S. 281–297. – URL https://projecteuclid.org/euclid.bsmsp/1200512992

[Mishali und Eldar 2010] Mishali, Moshe ; Eldar, Yonina C.: From theory to practice: Sub-Nyquist sampling of sparse wideband analog signals. In: *IEEE Journal of Selected Topics in Signal Processing* 4 (2010), Nr. 2, S. 375–391

[Nonaka und Da-te 1995] Nonaka, H. ; Da-te, Tsutomu: Ultrasonic position measurement and its applications to human interface. In: *IEEE Transactions on Instrumentation and Measurement* 44 (1995), Jun, Nr. 3, S. 771–774. – ISSN 0018-9456

[Press u. a. 1992] Press, William H. ; Teukolsky, Saul A. ; Vetterling, William T. ; Flannery, Brian P.: *Numerical recipes in Fortran 77: the art of scientific computing*. Bd. 2. Cambridge university press Cambridge, 1992. – URL http://www.nrbook.com/a/bookfpdf.html

[Raspberry Pi Foundation 2014] Raspberry Pi Foundation: *Raspberry Pi 3 Model B+*. : , 5 2014. – URL https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf. – Revision 1.1

[Reilly u. a. 2006] Reilly, SP ; Leach, RK ; Cuenat, A ; Awan, SA ; Lowe, M: Overview of MEMS sensors and the metrology requirements for their manufacture. In: *National Phsycial Laboratory* (2006)

[Savitzky und Golay 1964] Savitzky, Abraham ; Golay, Marcel J.: Smoothing and differentiation of data by simplified least squares procedures. In: *Analytical chemistry* 36 (1964), Nr. 8, S. 1627–1639

[Schmidt und Wuttke] Schmidt, Hans-Gottfried ; Wuttke, Frank: Seismische Standorterkundung für Anwendungen in der Geotechnik. In: *Messen in der Geotechnik*

[Steinbach u. a. 2000] Steinbach, Michael ; Karypis, George ; Kumar, Vipin u. a.: A comparison of document clustering techniques. In: *KDD workshop on text mining* Bd. 400 Boston (Veranst.), 2000, S. 525–526

[Suryanto 2006] Suryanto, Wiwit: *Rotational motions in seismology, theory and application*, Ph. D. thesis, Ph. D. dissertation, Dep. of Earth and Environ. Sci. Geophys., Univ. of Munich, Munich, Germany, Dissertation, 2006

[Thandu 2016] Thandu, Srinivas C.: *Algorithms leveraging smartphone sensing for analyzing explosion events*. Missouri University of Science and Technology, 2016

[Van Rossum und Drake Jr 1995] Van Rossum, Guido ; Drake Jr, Fred L.: *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995. – URL `www.python.org`

[Zhang 1998] Zhang, Gang: Design and Simulation of a CMOS-MEMS Accelerometer. In: *project report, Carnegie Mellon University, Pittsburgh, PA, May* (1998)

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 23. June 2018   Matthias Lösch