DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Thesis type (Bachelor's Thesis in Informatics, Master's Thesis in Robotics, . . .)

Thesis title

Author

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Thesis type (Bachelor's Thesis in Informatics, Master's Thesis in Robotics, . . .)

Thesis title

Titel der Abschlussarbeit

Author:	Author
Supervisor:	Supervisor
Advisor:	Advisor
Submission Date:	Submission date

I confirm that this thesis type (bachelor's thesis in informatics, master's thesis in robotics, ...) is my own work and I have documented all sources and material used.

Munich, Submission date   Author

Acknowledgments

Abstract

Contents

Acknowledgments **iii**

Abstract **iv**

1 Introduction **1**
1.1 Problem and solution overview 1
1.1.1 Motivation 1
1.1.2 Goals 1
1.2 Structure of the Thesis 2

2 Background **3**
2.1 Sensor 3
2.2 Sensor Fusion 5
2.3 Current Sensor Fusion Systems 6
2.4 Machine Learning 8
2.5 Deep Learning 8
2.5.1 Convulitional Neural Network 8
2.5.2 Back-propagation 9
2.5.3 Gradient based optimization for deep learning 9
2.5.4 Batch Normalization 9
2.6 Deep Learning 9
2.6.1 Activation Functions 10
2.6.2 Cost Functions 10
2.6.3 Back-propagation 10
2.6.4 Gradient based optimization for deep learning 10
2.7 Sensors 11
2.8 Inertial Mesurement Unit (IMU) 13
2.9 Sensorfutions 13

3 Relatedwork **15**

4 Simulation Setup **16**
4.1 Simulation 16
4.1.1 Simulation software 16
4.1.2 Scene Object 17
4.1.3 Inertial Measurement Units 18
4.1.3.1 Making IMU 19

5 Experiment Setup **21**
5.1 Simulation 21
5.1.1 Simulation software 21

6 Simulation **23**
6.1 Section 23
6.1.1 Subsection 23

7 Experiment Results and Comparison **25**
7.1 Section 25
7.1.1 Subsection 25

List of Figures **27**

List of Tables **28**

1 Introduction

Some Introduction will go here

1.1 Problem and solution overview

1.1.1 Motivation

In Augmented Reality or Mixed Reality, tracking is one of the most important as well as challenging tasks. Tracking means to detect the position and orientation of an object to a coordinate system in real-time. Tracking has to be very accurate, precise and robust otherwise misalignment will occur between the virtual and real objects in the frame which makes the experience much less pleasant to the user. In the sector of medical augmented reality, the situation can be severe.

There are many methods for tracking an object. We can track an object by multiple camera setup or use mechanical sensors such as Inertial Measurement Unit (IMU) or we can use hybrid approaches combining both camera and mechanical sensors.

The problem with tracking objects with cameras or hybrid systems is that it needs an external camera setup which is not portable and the setup can be problematic to the user. In order to track objects without the camera, we use a mechanical sensor such as IMU, which is portable and embedded in the Head Mounted Display (HMD). Here sensor fusion comes handy. Sensor fusion is a technic for combining sensory data from multiple sensors output which gives better results for tracking. In the case of IMU sensory data, prior work[] shows that increasing the number of IMU tends to provide better accuracy.

1.1.2 Goals

Our goal is to develop a deep learning model primarily with Convolutional Nural Network (CNN) and Dense Network to fuse multiple IMU data in a simulated environment. We will use many combinations of several IMUs and different CNN networks to predict the position and orientation of an object in a simulated environment and compare the results with other Deep Learning technics such as Recurrent Neural Network (RNN). We will take the combination which provides the best accuracy in the simulated environment and apply that model in a real-world scenario to test its accuracy to predict object pos.

Acceleration values from an IMU are well known to be highly unstable and there could be misalignment between axes. Using Deep Learning, we aim to achieve a stable and reliable position and orientation transformation without any camera, sensor calibration, registration, and prior error correction.

1.2 Structure of the Thesis

This document is divided in five chapters. In the second chapter, the necessary theoretical background is presented, whereas third chapter shows the structural solution and implementation. Chapter four introduces the obtained results and pertinent analysis, while the conclusions and proposed future work are summarized in chapter five. Afterwards, references and relevant bibliography are presented and the document ends with Appendices where outcomes of every experiment are detailed in their plots.

2 Background

2.1 Sensor

A sensor is a device that can monitor the environment that its designed for. Senson can be a single pice of hardware or can be a combination of hardware accuaring multiple information. Sensors vary in quality and price. Many modern sensors have on-board processing units to understand the data acquired from the sensing element without a separate computational platform. Remote sensing devices are sensors that can perceive information without physical contact. Unfortunately, sensors tend to have inherent problems including, but not limited to [Elmenreich, 2001]:

- Spatial coverage limits: Each sensor may only cover a certain region of space. For example, a dashboard camera will observe less surrounding region than a camera with a wide-view lens.
- Temporal coverage limits: Each sensor may only provide updates over certain periods of time, which may cause uncertainty between updates.
- Imprecision: Each sensor has limits to its sensing element. • Uncertainty: Unlike imprecision, uncertainty varies with the object being observed rather than the device making the observation. Uncertainty may be introduced by many environment factors or sensor defects in addition to time delay.
- Deprivation of sensor: Sensor element breakdown will cause loss of perception in environment. Many different types of sensors exist in the world, and each has its own unique ap plication. Four sensors are pertinent to the field of autonomous driving [Levinson et al., 2011]. The four sensors, their descriptions, uses, advantages and disadvantages are men tioned below:
- GPS: Global-positioning system (GPS) is a system of satellites and receivers used for global navigation of Earth designed by the U.S. military. GPS sends a signal to any GPS receiver with an unobstructed line of sight to four or more GPS satellites surrounding Earth [gps, 2011]. GPS is useful for finding the exact coordinates of a vehicle when it is in the line of sight of multiple satellites orbiting the Earth.

– Advantages: Precise coordinate measurements, fast, reliable in line of sight, externally-managed satellite systems. – Disadvantages: Expensive, subject to failure in bad weather conditions, subject to failure in distant locations where satellite coverage is blocked or unavailable, dependent on external data source, subject to hijacking and interference.

- Radar: RAdio Detection And Ranging (radar) is a remote sensing device that uses an antenna to scatter radio signals across a region in the direction it is pointing and listens for response signals that are reflected by objects in that area. Radar measures signal time of flight to determine the distance. Radars may use the doppler effect to compute speed based on shift in frequency of scattered waves as an object moves. Radar is useful for detecting obstacles, vehicles and pedestrians around a vehicle [Huang et al., 2016]. Tracking multiple targets at once is a primary use for an automotive radar. – Advantages: High-bandwidth signals, wide-spread area coverage, inde pendent from external systems, works in multiple weather conditions, light independent solution. – Disadvantages: Expensive, subject to interference, easy to corrupt signal with electromagnetic interference, many reflective radio responses make it harder to manage radar signals, algorithms for radar tracking are still imperfect, narrow field-of-view.

- Camera: A camera is an optical instrument that utilizes at least one converging or convex lens and a shutter to limit light intake into an enclosed housing for capturing images or recording image sequences [Kodak, 2017]. Video cameras work much like still-image cameras, but instead of simply capturing still images, they record a series of successive still images rapidly at a specific frame rate

  [Kodak, 2017]. A camera is useful for acquiring images or video sequences of object pixels in view of the lens in order to help detect, segment, and classify objects based on perceivable object properties like location, color, shape, edges and corners. – Advantages: Perceives high-level object characteristics like color, shape, and edges, perceives location relative to camera unit, easy to visualize data. – Disadvantages: Potential slow frame-rate update, image quality may be dependent on light, weather and various other factors, data-intensive processing, all cameras perceive objects differently, typically has a limited range of perception compared to other sensors, may be expensive.

- LiDAR: Light Detection and Ranging (LiDAR) is a method of remote sensing that uses light in the form of a pulsed laser to measure distance to an object based on signal time of flight [NOAA, 2012]. LiDAR is useful for perceiving surroundings when 3-dimensional, high-resolution, light-independent images are necessary. – Advantages: Independent of light, weather and external data sources, fast, accurate, 3-dimensional, high-resolution. – Disadvantages: Expensive, subject to interference by reflection or lack thereof, incompatible with transparent surfaces, data-intensive processing, less durable than other sensors.

2.2 Sensor Fusion

Sensor fusion is the act of combining data acquired from two or more sensors sources such that the resulting combination of sensory information provides a more certain description of factors observed by the separate sensors than would be if used individ ually [Elmenreich, 2001]. Sensor fusion is pertinent in many applications that entail the use of multiple sensors for inference and control. Examples of applications in clude intelligent and automated systems such as automotive driver assistance systems, autonomous robotics, and manufacturing robotics [Elmenreich, 2007]. Sensor fusion methods aim to solve many of the problems inherently present in sensors. Several important benefits may be derived from sensor fusion systems over single or disparate sensor sources. The benefits of sensor fusion over single source are the following [Elmenreich, 2001]:

- Reliability: Using multiple sensor sources introduces more resilience to partial sensor failure, which leads to greater redundancy and reliability.
- Extended spatial coverage: Each sensor may cover different areas. Combining the covered areas will lead to a greater overall coverage of surrounding environment and accommodate sensor deprivation.
- Extended temporal coverage: Each sensor may update at different time intervals, and thus interpolated sensor updates can be joined for increased tem poral coverage and decreased sensor deprivation.
- Increased Confidence: Combining sensor data will provide increased confidence by providing measurements resilient to the uncertainties in any particular sensor based on the combined coverage and error mitigation of all sensors.
- Reduced Uncertainty: Given the resilience of multiple sensors to the specific uncertainty of any one, the overall uncertainty of the perception system can be drastically reduced using sensor fusion.
- Robustness against noise: Multiple sensor sources can be used to determine when any one sensor has encountered noise in order to mitigate influence of noise in the system.

- Increased Resolution: Multiple sensor sources can be used to increase the resolution of measurements by combining all observations.

According to [Rao, 2001], sensor fusion can yield results that outperform the measurements of the single best sensor in the system if the fusion function class satisfies a proposed isolation property based on independently and identically distributed (iid) samples. The fusion function classes outlined that satisfy this isolation property are linear combinations, certain potential functions and feed-forward piecewise linear neural networks based on minimization of empirical error per sample.

2.3 Current Sensor Fusion Systems

Many researchers and companies have developed their own versions of sensor fusion systems for various purposes. Many of these systems are well-known an widely used in practice within the fields of automotive and robotics. In [Steux et al., 2002], a vehicle detection and tracking system using monocular color vision and radar data fusion using a 3-layer belief network was proposed called FADE. The fusion system focused on lower-level fusion and combined 12 different features to generate target position proposals each step and for each target. FADE performed in real-time and yielded good detection results in most cases according to scenarios recorded in a real car. A fusion system for collision warning using a single camera and radar was applied to detect and track vehicles in [Srinivasa et al., 2003]. The detections were fused using a probabilistic framework in order to compute reliable vehicle depth and azimuth angles. Their system clustered object detections into meta-tracks for each object and fused object tracks between the sensors. They found that the radar had many false positives due to multiple detections on large vehicles, structures, roadway signs and overhead structures. They also found that the camera had false positive detections on larger vehicles and roadway noise like potholes. Their system worked appropriately for nearby vehicles that were clearly visible by both sensors, but the system failed to detect vehicles more than 100 meters away due to insufficient resolution or vehicle occlusion. In [Dagan et al., 2004], engineers from Mobileye successfully applied a camera system to compute the time to collision (TTC) course from size and position of vehicles in the image. Although they did not test this theory, they mentioned the future use of radar and camera in a sensor fusion system since the radar would give more accurate range and range-rate measurements while the vision would solve angular accuracy problems of the radar. When the research was conducted, it was suggested that the fusion solution between radar and camera was costly, but since then, costs have decreased. A collision mitigation fusion system using a laser-scanner and stereo-vision was constructed and tested in [Labayrade et al., 2005]. The combination of the compli mentary laser scanner and stereo-vision sensors provided a high detection rate, low false alarm rate, and a system reactive to many obstacle occurrences. They men tioned that the laser-scanner was fast and accurate but could not be used alone due to many false alarms from collisions with the road surface and false detections with laser passes over obstacles. They also mentioned that stereo-vision was useful for modeling road geometry and obstacle detection, but it was not accurate for comput ing precise velocities or TTC for collision mitigation. In [Laneurit et al., 2003], a Kalman filter was successfully developed and applied for the purpose of sensor fusion between multiple sensors including GPS, wheel angle sensor, camera and LiDAR. They showed that this system was useful for detection and localization of vehicles on the road, especially when using the wheel angle sensor for detecting changes in vehicle direction. Their results revealed that cooperation between the positioning sensors for obstacle detection and location paired with LiDAR were able to improve global positioning of vehicles. A deep learning framework for signal estimation and classification applicable for mobile devices was created and tested in [Yao et al., 2016]. This framework applied convolutional and recurrent layers for regression and classification mobile comput ing tasks. The framework exploited local interactions of different sensing modalities using convolutional neural network (CNN)s, merged them into a global interaction and extracted temporal relationships via stacked GRU or LSTM layers. Their frame work achieved a notable mean absolute error on vehicle tracking regression tasks as compared to existing sensor fusion systems and high accuracy on human activity recognition classification tasks while it remained efficient enough to use on mobile devices like the Google Nexus 5 and Intel Edison. A multimodal, multi-stream deep learning framework designed to tackle the ego centric activity recognition using data fusion was proposed in [Song et al., 2016b]. To begin, they extended a multi-stream CNN to learn spatial and temporal features from egocentric videos. Then, they proposed a multi-stream LSTM architecture to learn features from multiple sensor streams including accelerometer and gyroscope. Third, they proposed a two-level fusion technique using SoftMax classification layers and dif ferent pooling methods to fuse the results of the neural networks in order to classify egocentric activities. The system performed worse than a hand-crafted multi-modal Fisher vector, but it was noted that hand-crafted features tended to perform better on smaller datasets. In review of the research, it seems there were limited amounts of data, flaws in the fusion design with SoftMax combination and flaws in the sensors, such as limited sensing capabilities. These factors all may have led to worse results than hand-crafted features on the utilized dataset. In [Wu et al., 2015], a multi-stream deep fusion neural network system using con volution neural networks and LSTM layers was applied to classify multi-modal temporal stream information in videos. Their adaptive multi-stream fusion system achieved an accuracy level much higher than other methods of fusion including averaging, ker nel averaging, multiple kernel learning (MKL), and logistic regression fusion methods.

2.4 Machine Learning

Machine Learning is a term that describes algorithms that learn from data. Goodfellow et al. quotes Mitchell from 1996 for a definition of what learning means, A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E. There is a wide variety of tasks T that can be done. For instance common tasks are regression, classification, anomaly detection, denoising and translation. Depending on what experience E they see during the learning process the most algorithms can be classified as supervised or unsupervised learning. Supervised learning algorithms experience a data set together with target values or labels that act as instructions on what to do. The unsupervised learning algorithms don't have any labels or targets but instead try to learn about the structure of the data set. The performance P can be how far off from the labels the output of the algorithm is in the case of supervised learning. This performance measure is often difficult to choose.

2.5 Deep Learning

Deep Learning is a term used for training deep artificial neural networks, or ANN s. An ANN is a function approximator consisting of multiple functions,

also called neurons. An ANN is called a feedforward ANN if there are no feedback connections of the output of the ANN being sent to itself. Each neuron is a linear function of some inputs, fed into a nonlinear function, called activation function. If the input of a neuron is a vector x, the output of the neuron gi can be written as

2.5.1 Convulitional Neural Network

The non-linear functions called activation functions, such as K in Equation 2.1, are typically fixed non-linear functions that are used to make the ANN able to approximate non-linear functions. If the non-linear activation functions weren't used, the output of the ANN would still be a linear function of the inputs x. In deep learning, there are a few commonly used functions that have become standard as activation functions. The rectifying linear unit, or ReLU is a function that is defined as

2.5.2 Back-propagation

When training a ANN to approximate a function, gradient based optimization is commonly used. To compute the gradient of a (loss) function f with respect to the parameters an algorithm called back-propagation is commonly used. It back-propagates from the objective function to gradients of the different weights and biases in the ANN to compute the gradient of the objective function with respect to all the parameters. If we have a function can be a cost function in a supervised learning setting but could also be other functions like a reward function in the reinforcement learning setting that will

2.5.3 Gradient based optimization for deep learning

The ANN weights and biases are updated to optimize some objective function with gradient based optimization, using the gradients computed with the backprop algorithm. The classic algorithm for optimizing the parameters in a ANN is called Gradient Descent. be an objective function that we want to minimize, where x is the input to the ANN the trainable parameters of the ANN. Then moving in the opposite direction of the gradient of L with respect to we move the parameters in a direction that makes the objective function smaller, which is what we want. However, usually x are random samples and the true gradient is the expected value of the gradient with respect to the random samples actually used. Thus when computing the gradient of the loss function as a function of some samples x, we are computing an unbiased noisy estimate of the gradient. This is referred to as Stochastic Gradient Descent, or SGD. Stochastic Gradient

2.5.4 Batch Normalization

Batch Normalization is a recent method in deep learning used to be able to train networks faster and use higher learning rates with decreased risk of divergence. It does so by making the normalization a part of the model architecture, fixing the mean and variances of the inputs to a layer by a normalization step. This makes the risk of the inputs to the activation functions getting in a range where the gradient vanishes smaller and allows for the use of higher learning rates.

2.6 Deep Learning

Deep Learning is a term used for training deep artificial neural networks, or ANN s. An ANN is a function approximator consisting of multiple functions, also called neurons. An ANN is called a feedforward ANN if there are no feedback connections of the output of the ANN being sent to itself. Each neuron is a linear function of some inputs, fed into a nonlinear function, called activation function. If the input of a neuron is a vector x, the output of the neuron gi can be written as

2.6.1 Activation Functions

The non-linear functions called activation functions, such as K in Equation 2.1, are typically fixed non-linear functions that are used to make the ANN able to approximate non-linear functions. If the non-linear activation functions weren't used, the output of the ANN would still be a linear function of the inputs x. In deep learning, there are a few commonly used functions that have become standard as activation functions. The rectifying linear unit, or ReLU is a function that is defined as

2.6.2 Cost Functions

To train a ANN, cost functions are usually minimized. In supervised learning, where we have labeled data to learn from, the cost functions are more straightforward than for instance in reinforcement learning that we will discuss in section The two main problems of supervised learning are regression and classification. In regression one wants to predict a numerical value whereas in classification the goal is to predict which class something belongs to given some inputs. Different loss functions are common for regression and classification. Most of them are however derived from the same principle, the one of Maximum Likelihood.

2.6.3 Back-propagation

When training a ANN to approximate a function, gradient based optimization is commonly used. To compute the gradient of a (loss) function f with respect to the parameters an algorithm called back-propagation is commonly used. It back-propagates from the objective function to gradients of the different weights and biases in the ANN to compute the gradient of the objective function with respect to all the parameters. If we have a function can be a cost function in a supervised learning setting but could also be other functions like a reward function in the reinforcement learning setting that will

2.6.4 Gradient based optimization for deep learning

The ANN weights and biases are updated to optimize some objective function with gradient based optimization, using the gradients computed with the backprop algorithm. The classic algorithm for optimizing the parameters in a ANN is called Gradient Descent. be an objective function that we want to minimize, where x is the input to the ANN the trainable parameters of the ANN. Then moving in the opposite direction of the gradient of L with respect to we move the parameters in a direction that makes the objective function smaller, which is what we want. However, usually x are random samples and the true gradient is the expected value of the gradient with respect to the random samples actually used. Thus when computing the gradient of the loss function as a function of some samples x, we are computing an unbiased noisy estimate of the gradient. This is referred to as Stochastic Gradient Descent, or SGD. Stochastic Gradient

2.7 Sensors

A sensor is a piece of hardware that monitors an environment based on a sensing element. Sensors vary in quality and price. Many modern sensors have on-board processing units to understand the data acquired from the sensing element without a separate computational platform. Remote sensing devices are sensors that can perceive information without physical contact. Unfortunately, sensors tend to have inherent problems including, but not limited to [Elmenreich, 2001]: • Spatial coverage limits: Each sensor may only cover a certain region of space. For example, a dashboard camera will observe less surrounding region than a camera with a wide-view lens. • Temporal coverage limits: Each sensor may only provide updates over certain periods of time, which may cause uncertainty between updates. • Imprecision: Each sensor has limits to its sensing element. • Uncertainty: Unlike imprecision, uncertainty varies with the object being observed rather than the device making the observation. Uncertainty may be introduced by many environment factors or sensor defects in addition to time delay. • Deprivation of sensor: Sensor element breakdown will cause loss of perception in environment. Many different types of sensors exist in the world, and each has its own unique ap plication. Four sensors are pertinent to the field of autonomous driving [Levinson et al., 2011]. The four sensors, their descriptions, uses, advantages and disadvantages are men tioned below: • GPS: Global-positioning system (GPS) is a system of satellites and receivers used for global navigation of Earth designed by the U.S. military. GPS sends a signal to any GPS receiver with an unobstructed line of sight to four or more GPS satellites surrounding Earth [gps, 2011]. GPS is useful for finding the exact coordinates of a vehicle when it is in the line of sight of multiple satellites orbiting the Earth.

– Advantages: Precise coordinate measurements, fast, reliable in line of sight, externally-managed satellite systems. – Disadvantages: Expensive, subject to failure in bad weather conditions, subject to failure in distant locations where satellite coverage is blocked or unavailable, dependent on external data source, subject to hijacking and interference. • Radar: RAdio Detection And Ranging (radar) is a remote sensing device that uses an antenna to scatter radio signals across a region in the direction it is pointing and listens for response signals that are reflected by objects in that area. Radar measures signal time of flight to determine the distance. Radars may use the doppler effect to compute speed based on shift in frequency of scattered waves as an object moves. Radar is useful for detecting obstacles, vehicles and pedestrians around a vehicle [Huang et al., 2016]. Tracking multiple targets at once is a primary use for an automotive radar. – Advantages: High-bandwidth signals, wide-spread area coverage, inde pendent from external systems, works in multiple weather conditions, light independent solution. – Disadvantages: Expensive, subject to interference, easy to corrupt signal with electromagnetic interference, many reflective radio responses make it harder to manage radar signals, algorithms for radar tracking are still imperfect, narrow field-of-view. • Camera: A camera is an optical instrument that utilizes at least one converging or convex lens and a shutter to limit light intake into an enclosed housing for capturing images or recording image sequences [Kodak, 2017]. Video cameras work much like still-image cameras, but instead of simply capturing still images, they record a series of successive still images rapidly at a specific frame rate

[Kodak, 2017]. A camera is useful for acquiring images or video sequences of object pixels in view of the lens in order to help detect, segment, and classify objects based on perceivable object properties like location, color, shape, edges and corners. – Advantages: Perceives high-level object characteristics like color, shape, and edges, perceives location relative to camera unit, easy to visualize data. – Disadvantages: Potential slow frame-rate update, image quality may be dependent on light, weather and various other factors, data-intensive processing, all cameras perceive objects differently, typically has a limited range of perception compared to other sensors, may be expensive. • LiDAR: Light Detection and Ranging (LiDAR) is a method of remote sensing that uses light in the form of a pulsed laser to measure distance to an object based on signal time of flight [NOAA, 2012]. LiDAR is useful for perceiving surroundings when 3-dimensional, high-resolution, light-independent images are necessary. – Advantages: Independent of light, weather and external data sources, fast, accurate, 3-dimensional, high-resolution. – Disadvantages: Expensive, subject to interference by reflection or lack thereof, incompatible with transparent surfaces, data-intensive processing, less durable than other sensors.

2.8 Inertial Mesurement Unit (IMU)

An inertial measurement unit is an electronic device that measures and reports a body's specific force, angular rate, and sometimes the orientation of the body, using a combination of accelerometers, gyroscopes, and sometimes magnetometers.

2.9 Sensorfutions

Sensor fusion is the act of combining data acquired from two or more sensors sources such that the resulting combination of sensory information provides a more certain description of factors observed by the separate sensors than would be if used individ ually [Elmenreich, 2001]. Sensor fusion is pertinent in many applications that entail the use of multiple sensors for inference and control. Examples of applications in clude intelligent and automated systems such as automotive driver assistance systems, autonomous robotics, and manufacturing robotics [Elmenreich, 2007]. Sensor fusion methods aim to solve many of the problems inherently present in sensors. Several important benefits may be derived from sensor fusion systems over single or disparate sensor sources. The benefits of sensor fusion over single source are the following [Elmenreich, 2001]:

• Reliability: Using multiple sensor sources introduces more resilience to partial sensor failure, which leads to greater redundancy and reliability. • Extended spatial coverage: Each sensor may cover different areas. Combining the covered areas will lead to a greater overall coverage of surrounding environment and accommodate sensor deprivation. • Extended temporal coverage: Each sensor may update at different time intervals, and thus interpolated sensor updates can be joined for increased tem poral coverage and decreased sensor deprivation. • Increased Confidence: Combining sensor data will provide increased confidence by providing measurements resilient to the uncertainties in any particular sensor based on the combined coverage and error mitigation of all sensors. • Reduced Uncertainty: Given the resilience of multiple sensors to the specific uncertainty of any one, the overall uncertainty of the perception system can be drastically reduced using sensor fusion. • Robustness against noise: Multiple sensor sources can be used to determine when any one sensor has encountered noise in order to mitigate influence of

noise in the system. • Increased Resolution: Multiple sensor sources can be used to increase the resolution of measurements by combining all observations. According to [Rao, 2001], sensor fusion can yield results that outperform the mea surements of the single best sensor in the system if the fusion function class satisfies a proposed isolation property based on independently and identically distributed (iid) samples. The fusion function classes outlined that satisfy this isolation property are linear combinations, certain potential functions and feed-forward piecewise linear neural networks based on minimization of empirical error per sample.

3 Relatedwork

4 Simulation Setup

The Experiment Setup can be divided into two phases, one is set up the simulation and collect data from the simulated setup. We call title this two-section as Simulation and Data Collection respectively. In this section we describe the detailed information for the simulation software and how we gathered data from the simulation and preprocessed the data to feed in the CNN algorithm.

We use simulated data because its easier to deal without any noise and doesn't need calibration and registration. The other advantages of the simulated data are that the simulated environment is ideal for this kind of experiment without any noisy data and its ideal for the analysis of the result before using it into the real world scenario and compare with it.

We obtained the from the simulation as .txt formate. then we processed the data and convert the data points in csv formate to use in our CNN model.

4.1 Simulation

As for the simulation we need to choose a simulation software that fulfills our requirement for simulating IMUs and moving the sensors on a flat plane on a predefined path. The other crucial requirements were, the data must be faultless and noise-free. For example, the item that had been utilized to mount the IMUs needed to move openly satisfying Six Degrees of Freedom (6DoF).

Moreover, the estimations from the virtual IMUs must be as exact as conceivable to imitate a real situation. We had fewer options for the simulation software to choose from. Among the simulation software we checked for our simulation MATLAB and CoppeliaSim Robotics were better performing.

4.1.1 Simulation software

After intensive investigation, we chose to go with the CoppeliaSim Robotics Education version for our simulation tool. Beforehand this product was known as V-Rep by Coppelia Robotics. One of the primary purposes behind picking this simulation tool is the assortment of accessible simple to utilize alternatives to mimic genuine situations. This product is created to recreate genuine situations for various mechanical parts for example Robot arms, Hexapods. It additionally has a wide scope of different segments for reproduction purposes. For instance – Infrastructure, furniture, family unit things, office things, and even people for reenactment purposes. This product offers a scope of usable virtual sensors for example – accelerometers, spinners, vision sensors, laser scanner, GPS sensors, and so forth.

The training adaptation is free for all. Since we didn't need to stress over the frequencies of various IMUs since every one of them is reproduced, the CPU recurrence made a difference the most while gathering the information. To analyze the reason, we ran the recreation on various CPU load. For example, with 4 IMUs and no other application running on the foundation, it took roughly 73 minutes to gather 1,06,437 information focuses. In a similar arrangement, however, with Google Chrome running on the foundation, the reenactment could just gather 47,509 information focuses on a similar time. It is required to run the reproduction remain solitary to get around the comparative number of information that focuses on a comparable time period.

In the following section we will describe the the components of the CoppeliaSim Robotics and how we used it to create the simulation

4.1.2 Scene Object

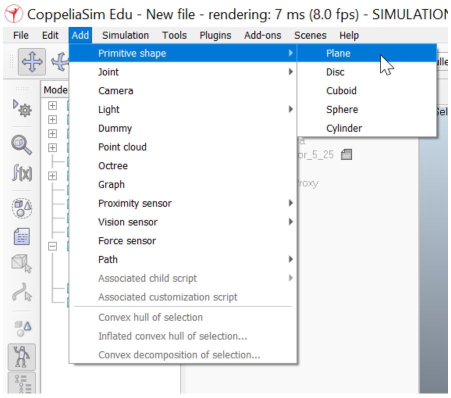
To recreate a real-world like situation, the main thing we required was an object for the scene. The object would represent the qualities of a moving segment in the simulated scenario

For our investigation, we required an object to contain the IMUs that would move along on a pre-characterized way. Obviously, 6 DoF must be guaranteed to make however much as could reasonably be expected accurate data. For our first methodology, we utilized effectively accessible predefined shapes that built-in the software. The software gives a number of predefined shapes such as plane, circle, cuboid, circle, and cylinder. We picked to utilize a two-dimensional plane as our object since it is the nearest to our model scenario.

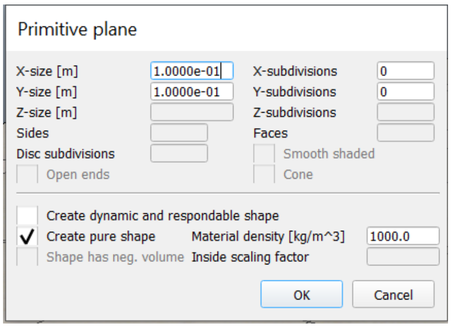
The plane is sufficiently large to contain at least 16 IMUs which is the most noteworthy number of IMUs we chose to utilize. The examples for putting the IMUs on the plane had been chosen together which would in the end help us to get the most ideal and solid information.

To add our ideal object to the scene, the following steps were :

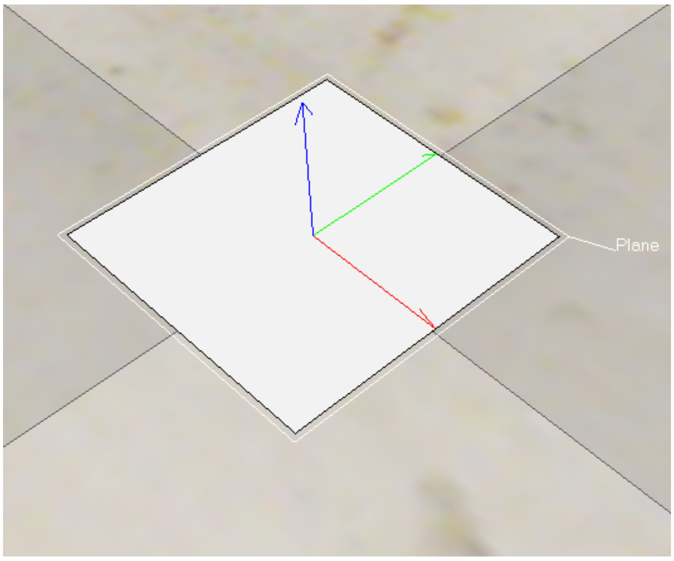
1. Click on 'Add' on the menu bar.
2. Go to the first option; 'Primitive Shape'.
3. Click on 'Plane' to add a two dimensional plane in our scene.



(a) 2D plane to the scene.



(b) Plane Configaration.



(c) Created 2D plane.

Figure 4.1: Creating scene object.

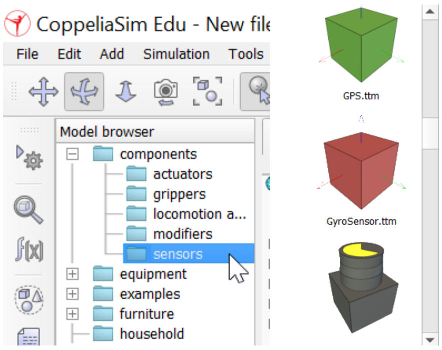
4. No change needed in the ‘Primitive plane’ dialog box and click ‘OK’.

4.1.3 Inertial Measurement Units

The next task is to add IMUs (Inertial Measurement Units) into our plane. The IMU containes an accelerometer and a gyroscope sensor. In the sensor section of the simulation tool, there is no IMU sensor out of the box but there are an accelerometer and a gyroscope sensor. To create IMU we combined these 2 sensors and added to to our plane.

To add one of those sensors to our scene, the following steps are needed to be followed:

1. Click on 'components' on the 'Model browser' pane.
2. Click on 'sensors'; this would open a new pane for all the available sensors just below the browser pane.
3. Scroll down the pane to select our desired sensors (accelerometer, gyroscope).
4. Drag and drop the sensors on our plane in the scene.



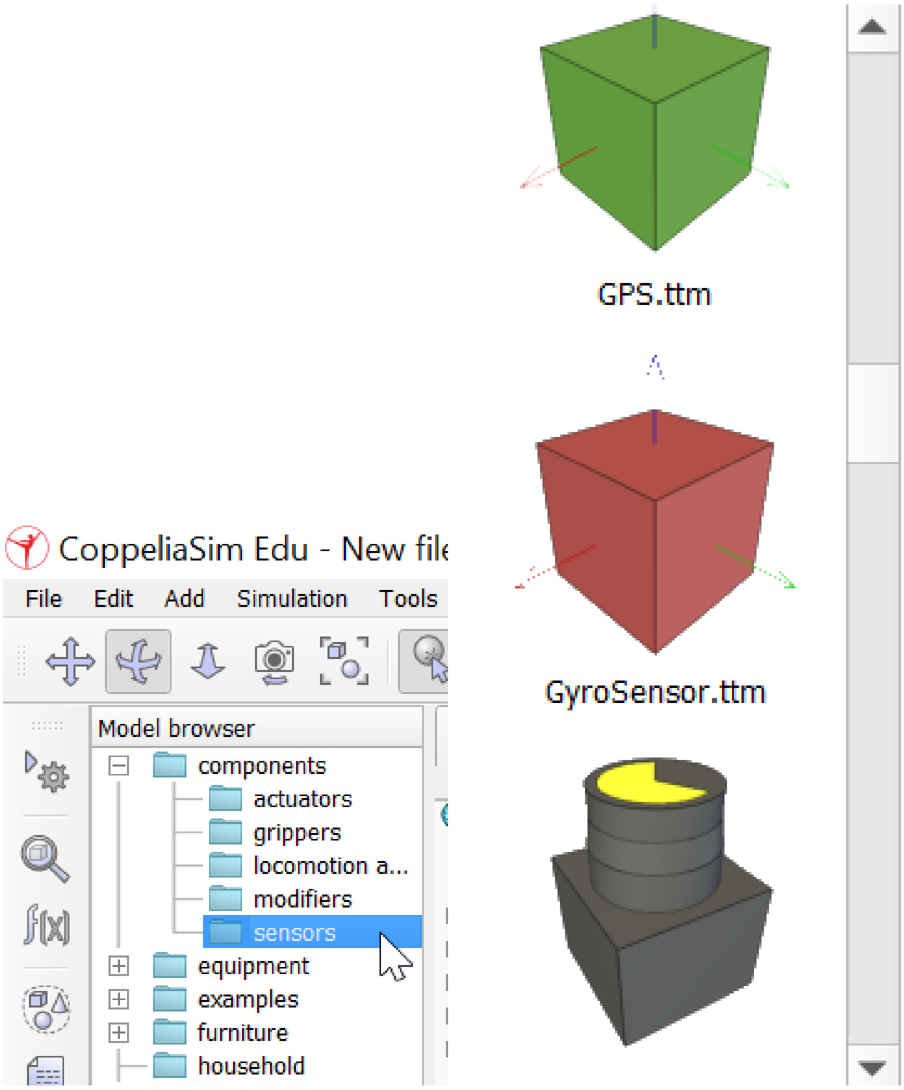
(a) selection sensor componant.

(b) selction accelerometer and a gyroscope sensor.

Figure 4.2: Creating IMU.

To combine the accelerometer and a gyroscope sensor and make a single IMU we need to change the LUA script attached with every sensor. Although it was possible just to drag and drop the sensors but we wanted to create a custom script to add a single unit IMU to incorporate our custom simulation control panel. we will discuss the control panel later on in the upcoming section.

4.1.3.1 Making IMU



(a) Coffee.

(b) More coffee.

Figure 4.3: The same cup of coffee. Two times.

5 Experiment Setup

The Experiment Setup can be divided into two phases, one is set up the simulation and collect data from the simulated setup. We call title this two-section as Simulation and Data Collection respectively. In this section we describe the detailed information for the simulation software and how we gathered data from the simulation and preprocessed the data to feed in the CNN algorithm. We use simulated data because its easier to deal without any noise and doesn't need calibration and registration. The other advantages of the simulated data are that the simulated environment is ideal for this kind of experiment without any noisy data and its ideal for the analysis of the result before using it into the real world scenario and compare with it.

We obtained the from the simulation as .txt formate. then we processed the data and convert the data points in csv formate to use in our CNN model.

5.1 Simulation

As for the simulation we need to choose a simulation software that fulfills our requirement for simulating IMUs and moving the sensors on a flat plane on a predefined path. The other crucial requirements were, the data must be faultless and noise-free. For example, the item that had been utilized to mount the IMUs needed to move openly satisfying Six Degrees of Freedom (6DoF). Moreover, the estimations from the virtual IMUs must be as exact as conceivable to imitate a real situation. We had fewer options for the simulation software to choose from. Among the simulation software we checked for our simulation MATLAB and CoppeliaSim Robotics were better performing.

5.1.1 Simulation software

After intensive investigation, we chose to go with the CoppeliaSim Robotics Education version for our simulation tool. Beforehand this product was known as V-Rep by Coppelia Robotics. One of the primary purposes behind picking this simulation tool is the assortment of accessible simple to utilize alternatives to mimic genuine situations. This product is created to recreate genuine situations for various mechanical parts for example Robot arms, Hexapods. It additionally has a wide scope of different segments for reproduction purposes. For instance – Infrastructure, furniture, family unit things, office things, and even people for reenactment purposes. This product offers a scope of usable virtual sensors for example – accelerometers, spinners, vision sensors, laser scanner, GPS sensors, and so forth. The training adaptation is free for all. Since we didn't need to stress over the frequencies of various IMUs since every one of them is reproduced, the CPU recurrence made a difference the most while gathering the information. The CPU must be liberated from use from different applications while the information assortment process is going on. To analyze the reason, we ran the recreation on various CPU load. For example, with 4 IMUs and no other application running on the foundation, it took roughly 73 minutes to gather 1,06,437 information focuses. In a similar arrangement, however, with Google Chrome running on the foundation, the reenactment could just gather 47,509 information focuses on a similar time. It is required to run the reproduction remain solitary to get around the comparative number of information that focuses on a comparable time period.

6 Simulation

6.1 Section

Citation test [**latex**].

6.1.1 Subsection

See Table 7.1, Figure 7.1, Figure 7.2, Figure 7.3.

Table 6.1: An example for a simple table.

A	B	C	D
1	2	1	2
2	3	2	3

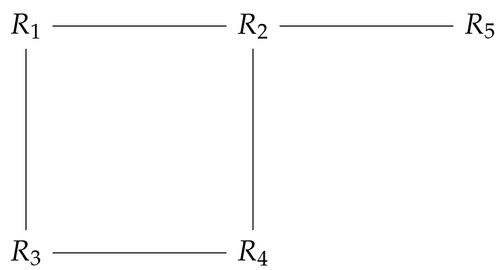


Figure 6.1: An example for a simple drawing.

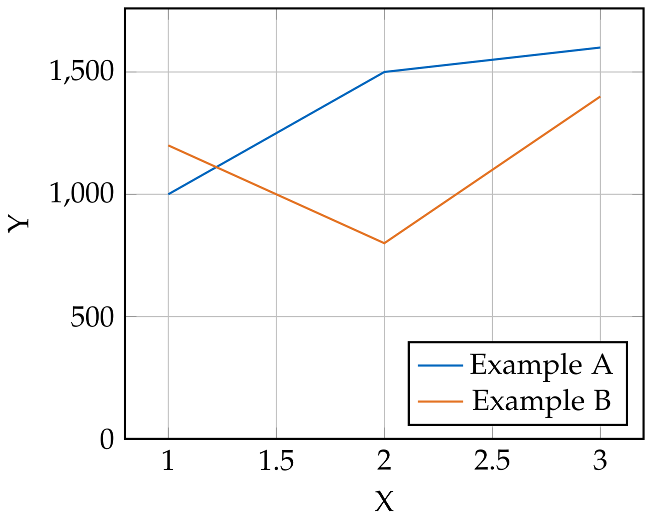


Figure 6.2: An example for a simple plot.

```
SELECT * FROM tbl WHERE tbl.str = "str"
```

Figure 6.3: An example for a source code listing.

7 Experiment Results and Comparison

7.1 Section

Citation test [**latex**].

7.1.1 Subsection

See Table 7.1, Figure 7.1, Figure 7.2, Figure 7.3.

Table 7.1: An example for a simple table.

A	B	C	D
1	2	1	2
2	3	2	3

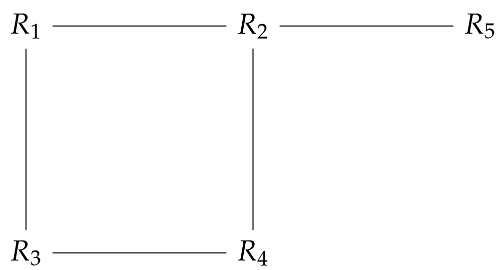


Figure 7.1: An example for a simple drawing.

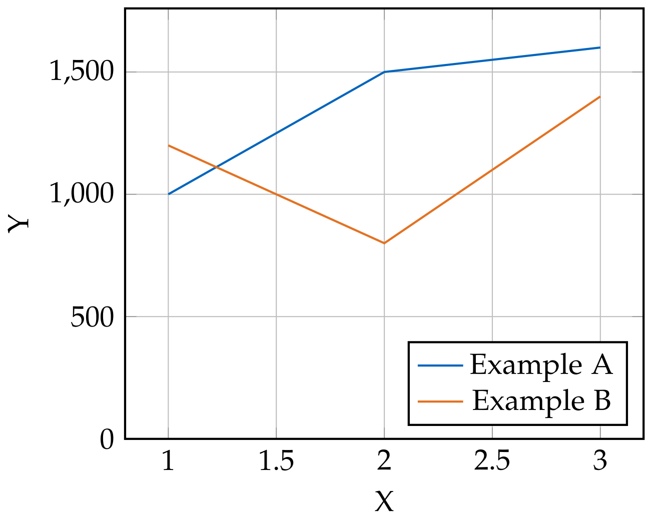


Figure 7.2: An example for a simple plot.

```
SELECT * FROM tbl WHERE tbl.str = "str"
```

Figure 7.3: An example for a source code listing.

List of Figures

4.1 Creating scene object. 18
4.2 Creating IMU. 19
4.3 The same cup of coffee. Two times. 20

6.1 Example drawing 23
6.2 Example plot 24
6.3 Example listing 24

7.1 Example drawing 25
7.2 Example plot 26
7.3 Example listing 26

List of Tables

6.1 Example table 23

7.1 Example table 25