Trajectory prediction of an ego-bike using IMU data from the cyclist's smartphone

Max Dellebeke

January 30, 2025

Abstract

Autonomous vehicles predict the trajectories of other road users to estimate future interferences. The autonomous vehicle might need to change its behaviour based on the future position of surrounding vehicles. This prediction is done using a large number of sensors. These sensors are used to collect data and predict trajectories based on these distant observations. Local data from these surrounding road users can help to establish a more accurate and advanced prediction. Data from Inertial Measurement Unit devices on vehicles is already used for trajectory predictions, but is not yet applied to bicycles in many occasions. Smartphones have an IMU device onboard that can gather data about directions, orientations, and movements. In this research, the applicability of the Robust Neural Inertial Navigation model to do trajectory estimations and predictions is investigated.

Keywords

Autonomous vehicles, trajectory prediction, Inertial Measurement Unit, Robust Neural Inertial Navigation, cycling behaviours

1 Introduction

Cycling comes with many health benefits, such as improved muscle strength, balance, heart health, and many more[8]. However, at the same time, cyclists are very vulnerable due to their lack of external protection. Therefore, cyclists are considered vulnerable road users and that is why it is important to improve their safety as much as possible[2]. In addition to the lack of protection, cyclist detection is one of the most difficult autonomous vehicle tasks, because of visual complexity, variety of appearances, and the lack of labelled datasets[5]. Hence, it is hard to determine the direction of bicycles, making it difficult to predict their future movements[9]. This increases the risk of accidents with cyclists and that is why improving the trajectory predictions of bicycles is important.

Autonomous vehicles use cameras and other sensors, like LiDAR to detect their environment[1]. The data that is gathered is used to predict the trajectories of the surrounding vehicles, pedestrians, and bicycles. Although these predictions are getting much better, they will always have to respond based on distant observations without the use data from bicycles themselves. To gather motion data from bicycles, an Inertial Measurement Unit (IMU) can be used. It consists of 3 accelerometers, 3 gyroscopes, and in some cases magnetometer sensors[12]. There is one for each axis per sensor. This device is used to determine the direction within a GPS system, track motions of phones and remotes, assist in aircraft manoeuvrers[4], and in autonomous vehicles.

2 Problem statement

Previous research shows that using a bicycle mounted IMU and camera is effective[6]. A big drawback of using bicycle mounted hardware is the scalability. It is costly to install sensors and cameras on bicycles and it does not add any features to the product. Therefore, in this research, the application of inertial navigation for bicycles is investigated. The motion of the upper leg can be tracked using the IMU in a smartphone when the cyclist carries the smartphone in his or her pocket. Almost every smartphone has an IMU that is used for activity tracking, health monitoring, navigating, and many other applications[16]. The scalability of using smartphones as a sensor is very high, since many people have such a device and carry them on their bicycles. Besides the high scalability, tracking the movements of the cyclist can have advantages over tracking the movements of the bicycle itself.

The RoNIN model is investigated to see if and how it can do trajectory predictions for bicycles. The model has shown to be very effective for indoor position and orientation estimations of moving subjects. It builds upon the robust IMU double integration (RIDI) and IONet models, where RoNIN adds neural architectures to improve on challenging motions. An important factor that makes this possible, is the new dataset that is collected. Instead of having a single device that collects IMU and VI-SLAM data in the RIDI and OXIOD (IONet) datasets, the RoNIN dataset uses two devices. One of the devices collects IMU, camera, and VI-SLAM data to generate an accurate ground truth, while the other device is the participant's smartphone that collects IMU data only and can be used in a usual manner.

In this paper, the ability to use the RoNIN model with a different dataset is investigated. The ego-bike dataset contains IMU data and unprocessed visual information only. The RoNIN model requires processed visual data from VI-SLAM software. In the RoNIN data collection, the Google Tango application finds pose information using the IMU and camera data[7]. Google Tango is an augmented reality computing platform that is able to do motion tracking and can find the orientation in 3D space[14]. To do this, it uses camera data in combination with IMU data to generate a vector that describes the pose of a participant. This information is used to find an accurate ground truth. The RoNIN dataset contains pose vectors and velocities that are not available in the ego-bike dataset. Although these vectors are missing, many of them can be composed from the IMU features. The features of the ego-bike dataset are extended to include these pose vectors. Then, the model can be trained and tested on the ego-bike dataset. Besides that, a detailed description is given about the RoNIN model to gain insights in its functionalities and requirements for other applications.

The model is analysed to see how it can be applied for this use case. The first research question is: *what features does the RoNIN model require to perform on the RoNIN dataset?*. To see if the ego-bike dataset fulfils these requirements, a sub research question is: *how can the ego-bike dataset be used with the RoNIN model?*.

3 Methodology

An existing dataset will be used that has data from participants cycling a route with a set of sensors attached to the bicycle. These sensors include a camera and three IMU devices. The IMU devices are mounted on the frame, the handlebar, and on one of the pedals. Although this dataset does not include an IMU device on the leg of the cyclist, the device that is mounted to the pedal follows a very similar movement. Therefore, this dataset can still be used for this research and a data collection phase is not required. For now, these two positions are considered similar enough, but in future research it is important to verify the results with data from an IMU in the pocket of the cyclist.

The behaviour of a cyclist is expected to have some relations to the future trajectory of the bike. Although there is barely any information available on body movements while cycling, there are some obvious assumptions. For example, many cyclists stop paddling to make a corner. A possible indicator for the direction of a corner could be the position of the legs or pedals. To make cornering easier, a cyclist can lower the outside pedal and apply pressure to be able to lean the bike to the direction of the corner[13]. A cyclist would then have its left leg, and therefore the left pedal, in the lower position to make a right corner and vice versa. These are assumptions based on my own experience and must therefore be researched further. The RoNIN model should be able to recognise these patterns to make predictions about the position and heading of the bicycle.

3.1 Model

Recurrent Neural Network, RNN in short, is a deep learning algorithm that uses the previous step as input for the current step stored in the memory state or hidden state[3]. Long Short-Term Memory, also known as LSTM, is an extension of RNN. LSTM has a memory cell that contains data for a longer period to learn long-term dependencies. The memory cell can store information that was gained previously but can also choose to forget if information becomes irrelevant[11].

The LSTM model has proven to be very effective in other prediction applications based on IMU data. For example, research done by Peng, Zhang, and Li [19] showed that this algorithm was able to predict a person falling 360 milliseconds ahead of the collision with 97,1% accuracy, 100,0% sensitivity, and 96,0% specificity. The accuracy for the trajectory prediction is expected to be lower than for the fall detection. That is because the movements are more similar during cycling and the prediction is about movements in a similar direction of where the cyclist is going. Although the accuracy could be high for prediction if the bicycle turns left or right, the accuracy for predicting the future position is expected to be much lower.

The Robust Neural Inertial Navigation, or RoNIN for short, is based on either ResNet, Long Short Term Memory Network (LSTM), or Temporal Convolutional Network (TCN)[21]. According to the results that are presented in the paper by Yan et al. (2019), the RoNIN model is the best performing model on the RoNIN dataset, which includes a lot of complex motions, compared to Naive Double Integration (NDI), Pedestrian Dead Reckoning (PDR), Robust IMU Double Integration (RIDI), and their own version of IONet. The RoNIN LSTM model has a slightly better performance than the RoNIN ResNet model on this dataset, but it takes much longer to train. Since the dataset that is used in this research contains a lot of complex motions as well, the RoNIN model with ResNet and LSTM as basis is likely to be the most suitable for this application.

3.2 Coordinate frame normalisation

The RoNIN model normalises the coordinate frame, which is important since the orientation of the IMU devices change continuously. Especially when data is used from a smartphone in the pocket of the cyclist. The RoNIN model uses a heading-agnostic coordinate frame (HACF) to represent the input and the output where the Z-axis is aligned with gravity [21]. Because the Z-axis is now normalised for all the frames, the data now describes the actual horizontal movements.

3.3 Velocity loss

The model finds a velocity for every IMU frame. This is achieved by calculating the derivative of low-frequency Visual Inertial and Simultaneous Localization And Mapping (VI-SLAM) poses. VI-SLAM is part of Project Tango [10]. First, the device captures images from the environment and looks for visual features [18]. When these features move around, they can be used to estimate the position of the camera by comparing the current location of these features to the location in previous frames. The visual information is able to provide a very accurate position, but requires a lot of processing power, while processing the IMU data is much more lightweight. Therefore, it fuses this visual information with IMU data to quickly find an accurate estimation. Since this VI-SLAM information is at a much lower frequency than the IMU data because of the low frequency of the camera, the velocity that it calculates is very noisy. Therefore, the RoNIN model applies two robust velocity losses. This results in a better signal-to-noise-ratio and better motion learning. The so called latent velocity loss is used for the RoNIN model with LSTM or TCN as backbone and it adds an integration layer that sums up 400 or 253 vectors respectively that are regressed by the model. Then, an L2 norm is applied to find the positional difference between the ground truth and the regressed window. This difference that is estimated by the sum of this window, must be equal to the positional difference in the ground truth. For the RoNIN ResNet model, it regresses positions instead of velocity vectors. Then, the mean squared error (MSE) is calculated to compare the regressed frame with the ground truth position.

4 Datasets

The RoNIN model comes with a large dataset containing many features, while the ego-bike dataset has less features. It is important to understand both datasets to be able to adjust either the model or the dataset accordingly. Both datasets are explained in the next two subsections and will help to get a better understanding of the applicability of the model and the possible required adjustments to the model or dataset.

4.1 Ego-bike dataset

The ego-bike dataset contains data from 32 participants cycling on a bicycle with various sensors. On this bicycle, a number of devices are installed that gather data from the trips. There are four IMU devices installed. One is mounted to the frame, one to the steering bar, one to the crank, and one to the helmet of the participant. The IMU device that is mounted to the crank is close to the pedal, but rotates with the crank. This means that the device is upside down at each rotation, instead of following the angle of the foot or pedal. There is also a GPS mounted to the frame as well as a camera to the steering bar. Each IMU provides data from its gyroscope and accelerometer in three axes measured at 200 Hz. The GPS provides the latitude and longitude coordinates, as well as velocity, acceleration in three axes, and gyroscope in three axes.

4.2 RoNIN dataset

The RoNIN dataset is 'the largest inertial navigation database consisting of more than 42.7 hours of IMU and ground truth 3D motion data from 100 human subjects.'[21]. The data is collected using two smartphones. The first one is a device that is used by the participants in a real life manner. The participants carry and use the phone like they would normally do. That means that this device can be carried in their hands being used while walking, be in a pocket and possibly be upside down, be put in a bag, etcetera. The orientation of the device is highly variable. Despite of this, the goal of the model is to estimate the position of the person. The features that are collected are the gyroscope, accelerometer, magnetometer, gravity, and pressure data from the IMU, as well as step count and device orientation (device_rv). The last two features are created by the smartphone's software using the IMU data. The second device is mounted to the chest of the participants using a harness. The data that is collected by this device is used to collect ground truth data. The same features are collected as the first device, but it also collects data that is created by the Tango platform.

In the first five seconds, the IMU of the chest-mounted smartphone and the IMU of the participant's smartphone are compared by walking in a straight line. The data from the chest-mounted smartphone is then used as the ground truth. The IMU of the participant's smartphone is adjusted by the offset angle, which is the angle between both smartphones.

4.3 Differences

The ego-bike dataset does not include any pose and velocity data from the Tango application, while the RoNIN dataset does have these features. The pose vectors and velocities provide a high quality ground truth for the model. There are three options to overcome this issue. The first option is to change the ground truth from using the Tango platform, to use a positional ground truth from the GPS. The trajectories can be constructed by combining the headings and velocities of every frame, but now that we have GPS data, the trajectory is already defined. Instead of finding a heading and velocity for every frame and comparing it against the one that is given by the Tango ground truth, the position of the device can be calculated and compared against the GPS ground truth instead. Since the GPS provides data at 10 Hz, the number of data points does not align with the IMU data that is collected at a rate of 200 Hz. The

RoNIN ResNet model makes predictions for every five IMU frames. Therefore, the GPS data must have at least 40 data points per second. For simplicity, the GPS data is interpolated to 200 Hz. Changing the model to use a positional ground truth was tried in this research, but due to the complexity of the RoNIN model and the time constraints, it was not possible to adjust the model and to find results.

A second option is to use VI-SLAM to create the pose vectors, velocities, and orientations (quaternions) using camera data. The ego-bike dataset does come with camera data that can be analysed by VI-SLAM. Google Tango is an application that can do exactly this, but it is developed for indoor navigation and is discontinued in 2017 to be replaced with their new augmented reality system called ARCore[15]. Although Tango proved to be a viable option for the RoNIN dataset, it might not be suitable for the dynamic outdoor environment. Song et al. [20] describes the so called DynaVINS, which is a robust VI-SLAM framework. It is applicable for dynamic environments and is able to detect features that are both dynamic and static. The model recognises dynamic features that cannot be used to find an estimation, while the static features are included.

The third option would be to create pose vectors, velocities, and orientations (quaternions) from the IMU data. These are the same features that are established by the Tango software as well. Because of that, the model does not need adjustments to work with a different type of ground truth. This option is tested in this research. The frame IMU is used for the ground truth, while the pedal IMU provides the input data. The gyroscope and accelerometer data can be combined into the device orientation and presented as quaternions. First of all, quaternions are established by integrating angular velocities over time. Then, the position is calculated by transforming the acceleration into the global frame using the quaternions and then integrating the global acceleration twice. The linear acceleration is then found by subtracting the gravity from the acceleration. After that, the gravity vector is estimated by rotating the quaternions into the global frame. Lastly, the game rotation vector is found by fusing accelerometer with gyroscope data. The ego-bike dataset does not provide any magnetometer, pressure, and step counting data. These are now the only features that cannot be added. For now, they are set to zero. That way, the model does not learn based on these features. The model is trained on the data of six different trips from three participants for 1000 epochs.

5 Results

After the model was trained, it was tested on a set of seen and unseen data. The results are presented in table 1 and 2. The model performed poorly on the provided dataset. The mean squared errors and angle errors are very high for both the seen and unseen data. During training, the loss was already high and the model converged very quickly. It is clear that the model does not learn well on the given dataset and is not able to provide results that are useful.

A number of problems could cause this poor performance. First of all, the pose vectors that are used for the ground truth will probably have drifted. In the RoNIN dataset, the VI-SLAM application is able to create a very high quality ground truth, because it can correct the position and pose based on visual features. The Tango application creates a 3D map and is able to find the position and orientation within that space resulting in a ground truth of high quality with minimal drift.

Second, the ego-bike dataset does not come with bias or calibration details. Without proper calibration and alignment, it is impossible to draw proper conclusions from the data. For example, calculating the linear acceleration requires accurate orientation estimations to be able to compensate for gravity properly. If the orientations are off, then the gravity is not aligned correctly with the Z-axis. This will result in improper coordinate frame normalisations. As described before, this is done using the heading-agnostic coordinate frame and is an important aspect of the model.

Third, not all the features that are missing from in the ego-bike dataset compared to the RoNIN dataset are filled. The missing pressure and step counting data is not likely to be much of an issue, but the missing magnetometer can affect the results signifi-

	MSE	Angle error
Data 1	0.9456649	1.513229
Data 2	1.0183111	1.592365
Data 3	0.86745447	1.3723118
Combined	0.94381016	1.4926354

Table 1: RoNIN body heading test results on seen data

	MSE	Angle error
Data 1	0.9481063	1.5153573
Data 2	1.0217904	1.5909898
Data 3	0.8415387	1.321282
Combined	0.9371452	1.4758763

Table 2: RoNIN body heading test results on unseen data

cantly. The magnetometer data can help to improve the quality and reliability of the features that are extracted from the accelerometer and gyroscope by using sensor fusion.

6 Future work

For future research it is useful to explore the options that are described in section 4.3. Most important are the options to improve the dataset. Using a VI-SLAM application to use the camera data will result in a much better ground truth. Besides that, the features that are extracted from the IMU can be improved using sensor fusion and the Extended Kalman Filter (EKF) as described by Laurell, Karlsson, and Naqqar [17]. In this research, just a small part of the ego-bike dataset could be used due to time constraints. But, improving the quality of the data and using a larger portion of the dataset will result in a model that performs much better.

7 Discussion

Trajectory predictions of bicycles is very complex because of visual complexity, variety of appearances, and the lack of labelled datasets. Yet it is a very important task for autonomous vehicles, since cyclists are very vulnerable. Models have been developed for indoor position estimations and predictions using inertial navigation. Applying such a model to bicycles could help to improve the trajectory predictions that are done by autonomous vehicles. The RoNIN model is an example of such a model for indoor inertial navigation. Applying this model to the ego-bike dataset has proven to be difficult. This dataset is lacking pose vectors, velocity, and orientations from a VI-SLAM application. Without it, the RoNIN model cannot define a proper ground truth that is required to train and test the model. There are three options to overcome this issue. The first option is to change the RoNIN model to only use the GPS as the ground truth. Then, another option is to use VI-SLIM to create these missing features. DynaVINS is a robust VI-SLAM framework that works in dynamic environments. The third option is to create the pose vectors, velocities, and orientations from the IMU data. This last option is explored in this research. The model was not able to provide good results on the dataset that was created using this method. Some of the features have been set to zero, which will affect the model negatively. Based on a magnetometer and step count of zero, the model expects that there is no movement in certain directions. The magnetometer should be added to the dataset and the step count would have to be removed from the RoNIN model, since the step count is irrelevant for bicycle trajectories. Recommendations for future research are given and include the use of VI-SLAM (DynaVINS) and the use of sensor fusion in combination with the EKF to improve the quality of the dataset.

References

[1] July 2022. URL: https://smartphotonics.nl/blog/how-lidar-works-in-autonomous-driving-vehicles-using-photonic-integrated-circuits/.

[2] Feb. 2023. URL: https://swov.nl/en/fact-sheet/cyclists.

[3] Dec. 2023. URL: https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network.

[4] Tarun Agarwal. *IMU sensor: Working Principle, usage and its applications*. Oct. 2019. URL: https://www.elprocus.com/imu-sensor-working-applications/.

[5] Mauricio Arias-Correa et al. "CYCLOPS: A cyclists' orientation data acquisition system using RGB camera and inertial measurement units (IMU)". In: *HardwareX* 18 (2024), e00534. ISSN: 2468-0672. DOI: https://doi.org/10.1016/j.ohx.2024.e00534. URL: https://www.sciencedirect.com/science/article/pii/S2468067224000282.

[6] Maouheb Bessi. *Trajectory Prediction on an ego-bike with IMU and camera using a Multi-State Constraint Kalman Filter*. 2023. URL: http://essay.utwente.nl/96435/.

[7] Rahil Bhagat. "Here's what you need to know about Google's project Tango". In: (June 2016). URL: https://www.forbes.com/sites/rahilbhagat/2016/06/20/just-what-is-googles-project-tango/.

[8] Lisa Brian. *8 surprising health benefits of cycling*. Apr. 2023. URL: https://www.goodhousekeeping.com/health/fitness/a43644224/cycling-benefits/.

[9] Peter Fairley. *Self-driving cars have a bicycle problem*. July 2021. URL: https://spectrum.ieee.org/selfdriving-cars-have-a-bicycle-problem.

[10] Taylor & Francis. *Project Tango - Knowledge and References — Taylor & Francis*. URL: https://taylorandfrancis.com/knowledge/Engineering_and_technology/Artificial_intelligence/Project_Tango/.

[11] GeeksforGeeks. *Deep learning: Introduction to long short term memory*. Dec. 2023. URL: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/.

[12] Laura Hayward. *Inertial Measurement Unit (IMU) - an introduction*. Jan. 2024. URL: https://www.advancednavigation.com/tech-articles/inertial-measurement-unit-imu-an-introduction/.

[13] *How to corner on a road bike — LIV Cycling*. URL: https://www.liv-cycling.com/global/campaigns/how-to-corner-on-a-road-bike/21772.

[14] Rabia Jafri et al. "Visual and Infrared Sensor Data-Based Obstacle Detection for the Visually Impaired Using the Google Project Tango Tablet Development Kit and the Unity Engine". In: *IEEE Access* 6 (2018), pp. 443–454. DOI: 10.1109/ACCESS.2017.2766579.

[15] Jacob Kastrenakes. "Google's Project Tango is shutting down because ARCore is already here". In: (Dec. 2017). URL: https://www.theverge.com/2017/12/15/16782556/project-tango-google-shutting-down-arcore-augmented-reality.

[16] Maan Khedr and Nasser El-Sheimy. "A Smartphone Step Counter Using IMU and Magnetometer for Navigation and Health Monitoring Applications". In: *Sensors* 17.11 (2017). ISSN: 1424-8220. DOI: 10.3390/s17112573. URL: https://www.mdpi.com/1424-8220/17/11/2573.

[17] Adam Laurell, Erik Karlsson, and Yousuf Naqqar. *GPS and IMU Sensor Fusion to Improve Velocity Accuracy*. 2022. URL: https://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-480286.

[18] Robert LiKamWa. *How Visual Inertial Odometry & SLAM Work*. June 2023. URL: https://sparks.learning.asu.edu/videos/how-visual-inertial-odometry-slam-work#:~:text=Visual%20inertial%20odometry%20(VIO)%20and,moving%20through%20a%20virtual%20environment.

[19] Jing Peng, Xianmin Zhang, and Hai Li. "A LSTM-Based Fall Prediction Method Using IMU". In: *Advances in Asian Mechanism and Machine Science*. Ed. by Nguyen Van Khang,

Nguyen Quang Hoang, and Marco Ceccarelli. Cham: Springer International Publishing, 2022, pp. 680–690. ISBN: 978-3-030-91892-7.

[20] Seungwon Song et al. "DynaVINS: A Visual-Inertial SLAM for Dynamic Environments". In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 11523–11530. DOI: 10.1109/LRA.2022.3203231.

[21] Hang Yan, Sachini Herath, and Yasutaka Furukawa. *RoNIN: Robust Neural Inertial Navigation in the Wild: Benchmark, Evaluations, and New Methods*. 2019. arXiv: 1905.12853 [cs.CV]. URL: https://arxiv.org/abs/1905.12853.