Drone-mounted RFID-based rack localization for assets in warehouses using deep learning

Jaron Fontaine*, Timo De Waele†, Adnan Shahid*, Emmeric Tanghe†, Pieter Suanel‡, Wout Joseph‡, Jeroen Hoebeke* and Eli De Poorter*

*imec - IDLab, Department of Information Technology at Ghent University
†imec - Waves, Department of Information Technology at Ghent University
‡ Aucxis CVBA, 9190 Stekene, Belgium.

Email: jaron.fontaine@ugent.be

Abstract—With the ongoing push towards an automated Industry 4.0, data-driven intelligent algorithms are getting more attention. Warehouse operators have traditionally required human labor to identify and register their assets. Autonomous flying drones will help alleviate this task by flying through the warehouse and detecting assets. This can be done based on vision, requiring expensive and energy consuming hardware, limiting drone flight time. In contrast, we propose a solution using radio-frequency identification (RFID) tags and machine learned algorithms to localize assets, which does not require a well-lit environment and can be processed in an energy efficient way. Our machine learning model achieves a 92-93 % accuracy, even when the drone is flying at different heights than the assets. Additionally, the model is easily implementable on off-the-shelf and low-energy consuming embedded hardware. This data-driven solution can easily be retrained for different environments and allows cheap RFID-based horizontal localization of assets in warehouses of the future.

Index Terms—Machine learning, localization, RFID

I. INTRODUCTION

One of the goals of Industry 4.0 is to further adopt intelligent technologies to assist humans in decision making and difficult tasks. As the global warehouse and storage market is expected to grow by 40% in the next 5 years, this industry proves to be an ideal testing ground for the integration of Industry 4.0 using smart technologies [1]. With the number and size of warehouses growing globally, scalable and reliable inventory management together with asset tracking becomes necessary. Traditionally, this has mostly been a manual task, where workers would scan each asset and update its location in a database. As warehouse racks commonly are higher than 8 meters, this task carries a high risk of injury. According to the U.S. Bureau of Labor Statistics Data almost 5% of warehouses workers experience a work related injury or illness each year [2]. Next to this relatively high risk of injury, the manual involvement for updating each asset’s location prevents real-time location information of the assets in the warehouse.

To this end, there has been a recent push in automating the process of asset tracking for inventory management. Automatic inventory management can be achieved through a variety of methods. Two prevalent technologies that are currently researched for this task are either autonomous robots that move over the ground or autonomous drones that fly between the corridors of the warehouses [3]. The benefit of using drones over ground-based robots is that they do not interfere with ground based operations such as forklifts and personnel. However, this approach also imposes some drawbacks, with the main ones being limited battery life and carrying capacity. Fortunately, these can be overcome by implementing landing pads with wireless charging and by equipping the drone with lightweight measuring equipment. For technologies that can perform asset tracking, there are again a couple of options available. They range from using QR-codes and computer vision to wireless technologies such as Bluetooth and RFID. Out of these technologies RFID shows to be the most promising, requiring no visible light and only using low-cost passive tags on each asset. However, as the RFID specification does not officially support localization, methods for performing this task have to be developed. One way of developing such an algorithm would be to investigate the received signal strength indicator (RSSI) or the phase of the received signal. This could give an estimation of the distance to the tag and the angle between the antenna and the tag.

Researchers have already managed to develop specialized algorithms that can estimate the 2D location of an asset on a rack to within 30 centimeters of accuracy using the received phase signal of a moving antenna [4]. However this approach required complex manual feature (MF) engineering which could impact the generalizability of the proposed approach. Another approach is to use precisely placed reference tags, for which the location is known. Here, researchers have managed to create algorithms that required little to no MF engineering to achieve sub 20 centimeters of accuracy [5] [6]. This comes, however, at the cost of a complex set-up phase in which the reference tags are placed, as well as an increase in time to perform each measurement, ranging from tens of seconds to a couple of minutes. Phase-Difference-of-Arrival (PDOA) between 2 or more antennas can be used to achieve similar or better accuracy, without the use of precisely placed reference tags with sub second reading times [7]. These approaches, however, require multiple antennas, making it less feasible for a drone base approach as this would severely reduce the performance and flight duration of the battery powered drone.

To solve the problems of MF engineering, the need for reference tags (which need to be placed on a static structure) and multiple antennas, we propose a convolutional neural
network (CNN) based deep learning (DL) approach that uses the RSSI and phase of the received signal to detect when a flying drone passes the horizontal location of an asset that is equipped with a passive RFID tag. The convolutional layers will be responsible for performing automatic feature extraction. The proposed approach requires no reference tags, works with only one antenna, can perform real-time inference and achieves a sub 35 cm accuracy. As such, the contribution in this paper is three fold. Firstly, we propose a DL model that can accurately and in real-time detect the position of an RFID tag, placed on an asset, with no MF engineering and requires only a single antenna. Secondly, we evaluate our approach on a dataset that was collected in a real-life warehouse instead of in a custom lab set-up. Thirdly, we evaluate our proposed solution on a cheap off-the-shelf embedded platform.

The rest of this paper will be structured as follows. In the next section, we will describe the data collection procedure, data characteristics and pre-processing, as well as the proposed DL model and the evaluation methodology. In Section III, we will discuss the obtained results and finally we will conclude the paper and give an overview of the future work in Section IV.

II. Approach

A. Data collection and description

Because drones have a limited weight they can carry, we investigate a solution which requires only one antenna. This is in contrast to solutions that make use of beam forming using antenna arrays and algorithms mathematically crafted by domain experts. In this paper, we intend to propose a DL solution that can perform equally well with a single antenna. To achieve a realistic dataset, we collected RFID tag responses using a drone which was flown by a professional drone pilot at near constant speeds and heights for each flight. Multiple speeds, ranging between 0.25 and 2 m/s, for different flights were executed to improve generalization within the dataset, especially for horizontal predictions. The drone collected the dataset in two corridors of a warehouse with racks at each side with a length of 19.6 m and a height of 9 m. In total, we used signals of 118 unique RFID tags, which have a minimum interval with neighbours tags of around 1 m horizontally and 2 m vertically, respectively. The collected signals exist of phase and RSSI samples, which were logged each time the antenna on the drone received a signal back from the tag. On average, a sampling frequency of around 20 Hz was achieved. All values are concatenated into a time series array. The result of one single data capture is shown in Figure 1a. The phase and RSSI both increase when the drone approaches a tag and decrease when it increases its distance again. However, finding the peak of both time series is not trivial as there are two main challenges: (i) The phase wraps around 360 degrees (or $2\pi$ radians), making the usage of traditional peak detection here, without additional filtering, not possible. One solution is to apply phase unwrapping, although this is still challenging with noisy data [8] (illustrated in Figure 1b), (ii) the RSSI is very sensitive to non-line-of-sight (NLOS) signal propagation, which is a real drawback in industrial environments like warehouses. As such, we propose a data-driven machine learning (ML) solution, which uses both phase and RSSI data simultaneously. The ground truth required to train our proposed CNN in a supervised fashion, was captured using a LIDAR sensor. Sampled at a frequency of 10 Hz, both the height of the drone and the distance to a fixed point at the end of the corridor was logged. This was later compared to the actual position of the tag to register the moment in time the drone flew by the tag for the horizontal ground truth label. Hence, this solution assumes that the position of the drone is available at all time, which is feasible e.g. using similar LIDAR systems or Ultra-wideband based localization systems.

We furthermore applied pre-processing of the collected dataset to filter out unwanted tags not relevant for the active flight, e.g. received signals from tags in other corridors. As such, we only used RFID signals with more than 50 samples. This is mostly the case for tags in the same corridor, on the same side of the antenna and for a height offset smaller than 2 meters. Additionally, the data for each run and each received unique tag is resampled into a time series of 1000 samples, for a consistent neural network input format.

B. Deep learning model for horizontal localization

In this paper, we introduce a ML model for horizontal asset localization. The proposed ML model is a CNN which requires only raw data, without any MF extraction. The convolutional layers are capable of learning and extracting features automatically, which makes model retraining or finetuning for different environments effortless.

Figure 2 presents the details of the CNN model, which is derived through careful experimentation and finetuning. The input layer is two dimensional and expects a 2D time series.
with phase and RSSI data. Next, a convolutional layer with 16 filters and a kernel size of 25 is attached. This layer makes use of kernel regularization, which penalizes updating too much layer parameters during optimizations, and is activated using the Rectified Linear Unit (ReLU) function. To reduce the complexity of the model, a MaxPooling layer of 10x1 follows next. Batch normalization is used, which will normalize the weights of the previous layer and can generally improve the predictive performance of the neural network. In order to prevent training data overfitting, a dropout of 20% is added. The neural network contains a second convolutional layer with the same settings as the first convolutional layer, but with 8 filters and a kernel size of 10. Similarly this layer is followed by a MaxPooling layer of 2x1, a batch normalization layer and a dropout layer of 20%. Next, a flatten layer is added and followed by a fully connected layer of 100 neurons. This layer is activated by the ReLU function, as is the next fully connected layer with 10 neurons. Finally, an output layer with one neuron and a linear activation function concludes the neural network. This output neuron predicts the position in the resampled time series, which is a location between 0 and 1000 in the input data, where the drone is passing the tag. This position can be mapped back to the original timestamp and thus horizontal position using the lidar time series data.

The CNN model is derived using supervised training by updating the weights using back-propagation. In particular, we optimized the weights with a batch size of 32, using the Adams gradient descent algorithm which minimizes any chosen loss function. Here, we opted for a mean absolute error (MAE) of 31.6 cm, presented in Table I. However, we can also observe that, even though the drone is flying at a different level, the ML algorithm still works well to identify the horizontal location of the tags, with minimal accuracy loss. Generally, a maximum error of around 150 cm is achieved, with a 90th percentile of < 75 cm.

As the goal of this paper is to obtain high accuracy in warehouse racks of 1 m width, achieving errors smaller than the other 80% of the dataset was used as training data. This ensures all data is evaluated, including potential outliers.

III. RESULTS

A. Accuracy

Figure 3 presents the cumulative distribution function (CDF) of the horizontal localization ML model. We differentiate 4 settings: (i) the drone height matches the height of the predicted tag location, (ii) the drone height does not match the height by flying at least two meters higher or lower than the considered tag, (iii) the combination of all of the previous two settings and finally (iv) a naïve positioning estimation method, without ML, by matching the LIDAR time series with the middle of the received RFID time series, where on average the tag is most likely to be. Overall, the highest accuracy is achieved when the drone is flying at the same level of the asset tag (with a mean absolute error (MAE) of 31.6 cm, presented in Table I). However, we can also observe that, even though the drone is flying at a different level, the ML algorithm still works well to identify the horizontal location of the tags, with minimal accuracy loss.

The ML models are trained using TensorFlow on a high-end NVIDIA V100 graphics processing unit (GPU), available in our in-house Virtual Wall [9]. All ML results in the next section are derived from 5-fold cross validation. This means that the dataset was split into 5 equal parts and the model was trained and evaluated 5 times. Each time, a unique 20% part of the dataset was used as unseen data for evaluation, while the other 80% of the dataset was used as training data. This ensures all data is evaluated, including potential outliers.
energy usage of 3.35 mJ per inference. To help alleviate manual labor and reduce risk of injury in warehouses, there is an upcoming need to automate warehouse inventory management. Drones using RFID antennas and assets with cheap passive RFID tags can provide a compelling solution. In this paper, we propose a ML solution for horizontal asset localization in real-life warehouse racks using only one antenna, no reference RFID tags and without the need for specialized algorithms that require time-consuming MF engineering. The proposed solution is a CNN approach, which uses RFID signals, i.e. RSSI and phase time series, together with LIDAR information and is easily implementable on embedded hardware (with 179 ms per inference). A mean absolute error of 34.5 cm is achieved, which allows a 93% accuracy for racks with 1 meter width, and 98% for racks with 2 meter width. We conclude that the solution, which can be effortlessly be retrained for other environments, enables cheap and accurate RFID-based horizontal localization of assets in warehouses.

We foresee future work in enabling vertical (height-based) prediction of assets, which could require e.g. data collection with omni-directional types of antennas and drones alternating heights during each flight. Additionally, generalization towards multiple environments needs to be analysed, where data augmentation techniques such as adding Gaussian noise, generating synthetic data through generative adversarial networks and transfer learning can be used.

ACKNOWLEDGMENTS

This work was supported by the Fund for Scientific Research Flanders, Belgium, FWO-Vlaanderen, FWO-SB, under Grant 1SB7619N, and by the imec.icon InWareDrones.

REFERENCES


### TABLE I: Mean absolute error and accuracy numbers for different rack widths

<table>
<thead>
<tr>
<th>Model</th>
<th>MAE</th>
<th>Accuracy (rack 1 m)</th>
<th>Accuracy (rack 2 m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drone height match</td>
<td>31.6 cm</td>
<td>93%</td>
<td>99%</td>
</tr>
<tr>
<td>No drone height match</td>
<td>38.0 cm</td>
<td>92%</td>
<td>98%</td>
</tr>
<tr>
<td>All heights</td>
<td>34.5 cm</td>
<td>93%</td>
<td>98%</td>
</tr>
<tr>
<td>All heights (with(int) only)</td>
<td>36.0 cm</td>
<td>92%</td>
<td>97%</td>
</tr>
<tr>
<td>All heights (RSSI only)</td>
<td>40.0 cm</td>
<td>90%</td>
<td>97%</td>
</tr>
<tr>
<td>Naive method</td>
<td>47.8 cm</td>
<td>57%</td>
<td>91%</td>
</tr>
</tbody>
</table>

50 cm is most important, because errors larger than 50 cm would start to overlap with neighbour shelves. This is further illustrated in Figure 4, where all data points of the horizontal localization errors are drawn. Here, points within the red lines, will not be confused with neighbouring rack shelves, considering a shelf width of 1 meter. When shelves of 2 meters in width are regarded, almost no misclassifications would occur. Table I presents the MAE and accuracy results when for racks of both 1 and 2 meter width. The ML algorithm produces a MAE of 34.5 cm over all heights, which corresponds to a classification accuracy of 93% and 98% when considering racks of 1 meter width and 2 m width, respectively. In order to investigate the impact of the combined input of phase and RSSI signals to the neural network, we have additionally evaluated each signal as input separately. Clearly, there is a significant advantage in combining the two signals, as using only the phase produces a MAE of 36.0 cm and using only the RSSI results in a MAE of 40.0 cm. In contrast, with the naïve positioning estimation method, without ML, accuracies of 57% and 91% are achieved, considering racks of 1 m width and 2 m width, respectively. This accuracy gap shows the importance of a good horizontal localization algorithm, especially in warehouses with narrow racks widths.

### B. Embedded inference and energy consumption

Finally for efficient communication, we investigate the feasibility of running the proposed ML algorithm on board of the drone. In this work, we considered a cheap and low-energy Arduino Nano 33 BLE with a popular Cortex M4 embedded processor. To enable embedded inference on this device, we converted the model into the TensorFlow Lite for Microcontrollers format. To optimize our model of 43321 parameters, we applied integer 8 bit quantization. This converts all parameters from float 32 bit to integer 8 bit precision. Without the embedded optimization, the model would not fit onto the memory of the device. With the embedded model we still obtain an accuracy of 35.2 cm, as compared to the original results of 34.5 cm for all heights. The feasibility of embedded ML is further highlighted by achieving a very low inference time of 179 ms and a memory footprint of 35.7 KB. This fast prediction performance translates into a low energy consumption overhead on the device. When measured with a Keysight N6705B DC power analyser, we measured a low energy usage of 3.35 mJ per inference.