



---

# Web Service for Point Cloud Supported Robot Programming Using Machine Learning

Kristina Enes

---

## Abstract

In industrial automation, the use of robots is already standard. But there is still a lot of room for further automation. One such place where improvements can be made is in the adjustment of a production system to new and unknown products. Currently, this task includes the reprogramming of the robot and a readjustment of the image processing algorithms if sensors are involved. This takes time, effort, and a specialist, something especially small and middle-sized companies shy away from. We propose to represent a physical production line with a digital twin, using the simulated production system to generate labeled data to be used for training in a deep learning component. An artificial neural network will be trained to both recognize and localize the observed products. This allows the production line to handle both known and unknown products more flexible. The deep learning component itself is located in a cloud and can be accessed through a web service, allowing any member of the staff to initiate the training, regardless of their programming skills. In summary, our approach addresses not only further automation in manufacturing but also the use of synthesized data for deep learning.

---

## Keywords

Machine learning · Digital twin · Structured-light 3D scanner

---

K. Enes (✉)

RIF Institut Für Forschung Und Transfer E.V, Joseph-von-Fraunhofer Str. 20, 44227 Dortmund, Germany

e-mail: [Kristina.Enes@rt.rif-ev.de](mailto:Kristina.Enes@rt.rif-ev.de)

© The Author(s) 2022

T. Schüppstuhl et al. (eds.), *Annals of Scientific Society for Assembly, Handling and Industrial Robotics 2021*,

[https://doi.org/10.1007/978-3-030-74032-0\\_21](https://doi.org/10.1007/978-3-030-74032-0_21)

## 1 Introduction

This paper addresses three issues regarding machine learning in the industrial field.

Firstly, many problems can arise when applying machine learning in a production process. Artificial neural networks need a large amount of data to train on and while there is a lot available containing day to day situations, like indoor and outdoor scenes, this is not the case for industrial environments. Suitable training data need to be collected and labeled manually, which can be difficult and tedious. Another problem is that the environment can influence the quality of the data, like distortions due to vibrations or due to changes in temperature.

Secondly, while the production of large lot sizes is mostly automated, the demand for small lot sizes increases, as well as the variety of the products. Also, the number of times the production process needs to be adjusted to new products increases, too. Since both the reprogramming of the robots, that are included in the process, as well as the image recognition algorithms, are tailored to the environment and the products, the incorporation of a new product to the production process is costly. An expert with programming skills is required. But such experts are not always available.

And thirdly, the acquisition of suitable training data for a specific task. Though there are many datasets with large amounts of training data they do not cover every situation. There are many environments in which training data is nearly impossible to obtain, e.g. the deep sea or space. It is also hard to gather data on situations that rarely occur naturally. Creating your own dataset of real-world data will take time and effort, since besides gathering the data, they need to be labeled manually, too.

Our approach relates to all three challenges. We propose creating a digital twin mirroring a physical production line in the industry to generate synthesized training data to train an artificial neural network, accessible through a web service, to learn to recognize and locate products given observed point clouds of the scene.

In a simulation, we can generate an unlimited amount of training data and the labeling of the data is automated. Also, a simulation can easily show extreme situations that rarely occur. With this we can generate a wide range of training data, giving a possible solution to the problem of insufficient real-world training data.

If an artificial neural network can successfully be trained with synthetic data with or without a smaller amount of real-world data to recognize and locate products, the flexibility of the production process increases, and the costs decrease. A new product can easily be incorporated by every member of the staff without the need of any programming skills and without the need to consult a specialist. With this, producing small lot sizes of a large variety will become more efficient.

Our solution is also one step in the direction of introducing machine learning into the industry, showing that the issues with insufficient data and error-prone sensor data can be approached by using simulations.

The following parts of this paper are structured as follows: in Sect. 2 we describe some previous work on the above topics. In Sect. 3 we introduce some works which will be used as a basis for our solution, VEROSIM, and Rüstflex. The architecture and components of our system are described in Sect. 4. We will discuss further plans for our project in Sect. 5.

---

## 2 Related Work

Rossano et al. [12] describe how specialized knowledge is needed to program industrial robots. They give an overview of approaches to help with the program structure and the creation of new motion paths. The solutions suggested are either graphic-based, CAD-based, or include manually moving the robot arm. But mostly those solutions have drawbacks. A current approach used by Drag & Bot [5] or ArtiMinds [3] includes the use of function blocks in the form of graphic program modules. In IntellAct [16] a robot learns by observing a human manually demonstrating certain tasks.

Sahbani et al. [15] describe two different basic research approaches for a robot to grip unknown objects. Firstly, the analytical approach, which is based on a mathematical model. Secondly, the empirical approach, which includes the imitation of human motions or planning a motion based on observations of an object. Bohg et al. [4] give an overview of the data-based planning of gripping an object. However, the approach of using deep learning in this context is relatively new.

Digital twins are researched since approximately 2010. Negri et al. [9] give a survey of current research regarding digital twins. They define digital twins, in the context of production systems, as a virtual representation of a production process that can be used for different types of simulations. Rossmann and Schluse [13] define experimental digital twins, which can be combined into complex simulation models, namely virtual testbeds. They represent all important parts of an application close to reality and can be used for interactive experiments.

Georgakis et al. [7] synthesize existing real-world training data of indoor scenes by superimposing images. Textured object models are placed into different background scenes at different locations and with different sizes. This is done by either image blending or by using depth and semantic information to make smart positioning. An object detector was trained using these images in addition to an existing dataset with good results. This approach allows the expansion of existing datasets, but in case of applications with no suitable dataset, a different solution is needed.

Tsai et al. [17] create a large set of computer-generated 3D hand images to train a convolutional neural network to identify different hand gestures. They discovered that adding about 0.09% of real-world images to the training process increases the accuracy from 37.5 to 77.08%. Lindgren et al. [8] generate a dataset of synthetic hand gestures by using modern gaming engines. The synthetic hand gestures are created by making

variations to the kinematics. They train a classifier purely on those generated data. The results are accurate and can be used on real-world data.

Richter et al. [11] use modern computer games to generate labeled training data for semantic segmentation tasks. They add different amounts of synthetic data to two different semantic segmentation datasets and compare the results of the trained networks. They have shown that by including synthetic data to real-world data the performance of the network increases, reducing the amount of hand-labeled data needed to train a successful neural network. But in this case, the generated data is completely dependent on the game. Using this approach for a specific application might not be possible, since the influence, the user has on the resulting training data, is restricted and there might not be a game suitable for your application.

---

## 3 Groundwork

### 3.1 Rüstflex

Rüstflex [14] is a web application created by Vathos GmbH [18] which can efficiently retool industry robots. It can adjust a robot's movements, mostly those that can be executed without sensory aid. The application runs either in a cloud or in a local computer center and can be accessed through all mobile end devices. It provides a formula where all relevant information for the setup of an article is stored. We will use Rüstflex in our project for the parametrization of the demonstrator (see Sect. 4.5) to reprogram the robot.

### 3.2 VEROSIM

With our simulation framework VEROSIM [19] we can create digital twins and virtual testbeds. A virtual testbed provides us with a completely virtual environment that can be used for experimentation. It can be integrated into real systems and can provide intelligent sensors, actors, and robots. We have already created a wide range of applications ranging from those in the industry to natural and urban environments as well as space. Three example applications are shown in Fig. 1. Our framework also provides several sensor simulations like ToF cameras, laser scanners, or radars, which can be built upon. One of the goals of this project is to upgrade our sensor simulations.

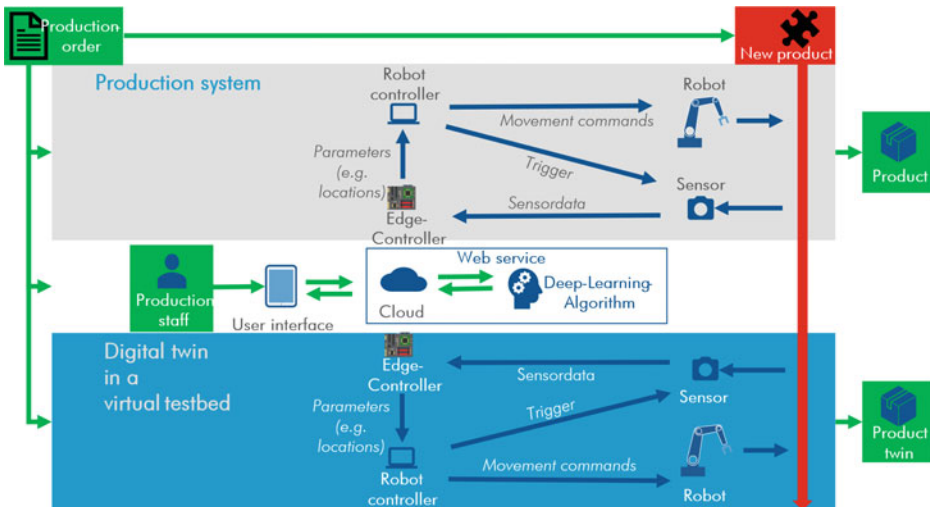
We will use VEROSIM in our research project to create digital twins of a production line and all its components, as well as to generate synthetic and automatically labeled data to be used as training data for an artificial neural network.



**Fig. 1** Digital twins and virtual testbeds in industry, forestry, and space

### 4 System Components

Our goal is to enable a robot to handle unknown objects using machine learning methods trained mainly on synthetic data. Our overall system consists of several parts. First, there is a physical production line and its digital twin. A digital twin simulates all parts of its real-world counterpart. It can consist of several other digital twins and simulates the communication streams between different components. A deep learning component located in a cloud can be accessed by the physical production system and the digital twin as well as a user through a web service. All these components will be combined in a demonstrator. The overall architecture of our system is shown in Fig. 2. Further descriptions of each component are given in the following chapters.



**Fig. 2** System architecture

## 4.1 Production Line

We view a physical production process as the basis of our system. This process consists of a pick and place robot, a sensor, a gripper, a robot controller, an edge controller, and products of the production line. The robot controller dictates the robot's movements, opens, and closes the gripper, and triggers the sensor. The edge controller contains a copy of an artificial neural network trained in a cloud. The deep learning component is further described in Sect. 4.3.

The edge controller receives the data generated by the sensor and gives them as input to the copy of the neural network. In this case the edge controller returns the positions of the products to the robot controller.

The specific production line we chose is situated at *aha! Albert Haag GmbH* [1]. It is a deep drawing process where products are redrawn one or several times. Between each redrawing a robot from type *Fanuc* [6] takes and places the processed product from a machine to a palette with a piece of cardboard between each layer of products. Then a new product is taken and placed from a palette to the machine to be redrawn again.

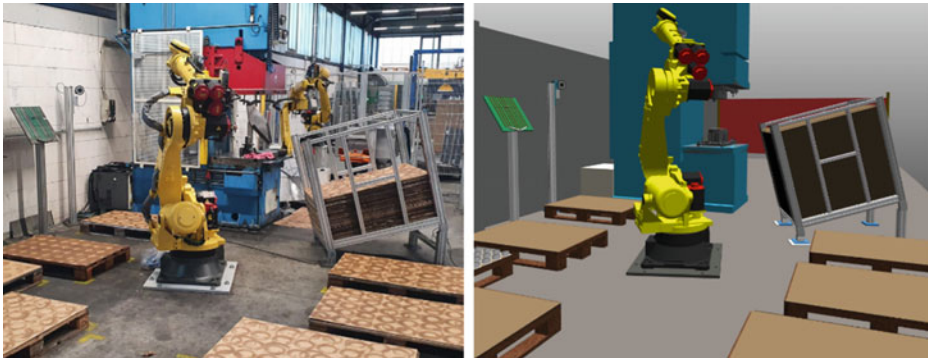
There are many different variations of products that can be processed in this production line. They differ in size, material, geometry, and the number of required deep drawings. Also, each product is oiled. The sensor we use should be able to make accurate recordings of these products independent of their material or shininess. The observed data is then used to recognize and locate the products in the topmost layer of the current palette. The sensor we choose is a structured-light 3D scanner. It can produce both images and point clouds. The resulting point clouds are forwarded to the edge controller.

## 4.2 Digital Twin

Given the physical production line we build its digital twin using our simulation framework *VEROSIM*. But to what extent does a digital twin resemble its counterpart. Here are some, but not all, examples of what exactly is contained in a digital twin and what it can do:

- It can have physical attributes like geometry, material, and texture.
- It can manage working data, which is generated during its application.
- It can execute different functions and services.
- It contains interfaces for communication.

We create a digital twin of all components of the production process while keeping the above points and more in mind. For the robot we define the kinematics and inverse kinematics. The robot controller can move the robot by following a predefined path and the sensor generates synthesized point clouds of the observed scene. The difficulty regarding the simulation of the sensor are the sensor's internal and external errors that occur during the recording. Our sensor simulation framework should be able to simulate



**Fig. 3** Production process from aha! Albert Haag GmbH [1] and its digital twin

those errors, too, since our goal is to generate a simulation that is as close to reality as possible. But since the simulation contains all important information it is easy to automatically label all generated data.

In Fig. 3 you can see a picture of the physical production line on the left and its digital twin on the right.

### 4.3 Deep Learning

Both the physical production line as well as the simulated production line are capable of exchanging data through the physical and simulated edge controller with a deep learning component located in a cloud. The physical edge controller can load a copy of a trained neural network and provides a method for local inference. On the other hand, the simulated edge controller can send generated data to the cloud as training data or it can send unlabeled data for inference. In the latter case predicted parameters describing the class and the locations of the currently observed products are returned.

The deep learning component uses the training data, generated by the digital twin and the physical production system, to train a neural network to both recognize and locate new products. It detects objects in point cloud recordings of a scene. The training data we need to generate consists of three parts. Firstly, the point cloud of a scene. Secondly all visible bounding boxes in this scene and thirdly a mapping from each point in the point cloud to the center of the corresponding bounding box. The deep learning component and the webservice will be provided by Vathos GmbH.

### 4.4 Web Service

For easy access to the cloud and the deep learning component a web service is used. The web service allows a user in the production staff to upload the CAD data of new and

unknown products to the cloud. He should also be able to initiate the start of the training process since this is the most expensive part of our system. The physical edge controller can synchronize with the cloud, loading a copy of the current neural network for a local usage. In this case the production system will keep working even during disturbances in the internet connection. Besides, the time needed for inference is shorter by using the edge controller in contrast to the time needed to access the cloud for inference. The simulated edge controller has a different duty than the physical one. Through the web service it can download the CAD data of the new products and replace old products with the new one in our simulation. After generating labeled training data, the edge controller can use the web service to upload them to the cloud where a neural net will use them to train.

Since Rüstflex also uses a web service for easy usage, it can be used as groundwork for the web service used in our project, which is currently in progress.

## 4.5 Demonstrator

The demonstrator is based on the production process from aha! Albert Haag GmbH. It combines all previously mentioned components. A simplified version of the described production line will be built by Arthur Bräuer GmbH & Co. KG [2], an integrator of robot arrangements. The goal of the demonstrator is to show that it is possible to easily adjust an automated process to new and unknown products using only data driven algorithms. We will use it to test our system in the applications from both aha! Albert Haag GmbH as well as Rewe Digital [10]. While the process from aha! Albert Haag GmbH is used to build our system on, a different process from Rewe Digital in logistics is considered to ensure the generalization of our system. For the parametrization of the robot, we will use Rüstflex. The results of the demonstrator will then be used to optimize the other components of our system.

---

## 5 Conclusions and Future Work

We are currently in the beginning stages of this project. Until now we have worked out the finer details of our system, specified the system architecture, which is partly shown in Fig. 2. We have specified the workflow of our chosen production line and built a first simulation model of said process. Vathos GmbH is developing the web service and the deep learning component. Currently we work on a digital twin of our chosen sensor and a component for labeling the generated data.

Further plans regarding this project contain finalizing our simulated sensor and training a network using the resulting data. At the end of our project, we hope to show that it is possible to improve a production process using our system and applying machine learning in industry using only synthetic data. Since a lot of clients wish for their production lines to be capable of dealing with small lot sizes, our approach would enable Bräuer GmbH &



Co. KG to offer their clients more flexible production systems. In addition, small and middle-sized companies like Rewe Digital and aha! Albert Haag GmbH could improve their current and future production systems. We hope for our results to help many other such companies to further automate their manufacturing processes.

And lastly, the success of our project will lead to the capability of industry robots to solve certain tasks more autonomously, making it possible to abstract the instructions given to the robot.

---

## References

1. aha! Albert Haag GmbH (2020). <https://www.aha-haag.de/>. Accessed 10 Aug 2020
2. Arthur Bräuer GmbH & Co. KG (2020). <https://www.braeuergmbh.de/>. Accessed 10 Aug 2020
3. Artiminds (2020). <https://www.artiminds.com>. Accessed 14 Aug 2020
4. Bohg, J., Morales, A., Asfour, T., Kragic, D.: Data-driven grasp synthesis a survey. *IEEE Trans. Rob.* **30**(2), 289–309 (2014)
5. Drag & Bot (2020). <https://www.dragandbot.com/product/>. Accessed 14 Aug 2020
6. Fanuc Robots (2020). <https://www.fanuc.eu/de/en>. Accessed 10 Aug 2020
7. Georgakis, G., Mousavian, A., Berg, A.C., Kosecka, J.: Synthesizing Training Data for Object Detection in Indoor Scenes (2017)
8. Lindgren, K., Kalavakonda, N., Caballero, D.En., Huang, K., Hannaford, B.: Learned hand gesture classification through synthetically generated training samples. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3937–3942 (2018)
9. Negri, E., Fumagalli, L., Macchi, M.: A review of the roles of digital twin in CPS-based production systems. *Proc. Manuf.* **11**, 939–948 (2017)
10. Rewe Digital (2020). <https://www.rewe-digital.com/>. Accessed 12 Aug 2020
11. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: ground truth from computer games. In: European Conference on Computer Vision (ECCV), vol. 9906, pp. 102–118. Springer (2016)
12. Rossano, G.F., Martinez, C., Hedelind, M., Murphy, S., Fuhlbrigge, T.A.: Easy robot programming concepts: an industrial perspective. In: 2013 IEEE International Conference on Automation Science and Engineering (CASE), pp. 1119–1126, Madison, Wisconsin, USA (2013)
13. Rossmann, J., Schluse, M.: Virtual robotic testbeds: a foundation for e-robotics in space, in industry—and in the woods. In: *Developments in E-Systems Engineering*, pp. 496–501, Dubai (2011)
14. Rüstflex (2020). <https://uniktec.de/ruestflex.php>. Accessed 10 Aug 2020
15. Sahbani, A., El-Khoury, S., Bidaud, P.: An overview of 3D object grasp synthesis algorithms. *Robot. Auton. Syst.* **60**(3), 326–336 (2012)
16. Savarimuthu, T.R., Buch, A.G., Schlette, C., Wantia, N., Roßmann, J., Martinez, D., Alenyà, G., Torras, C., Ude, A., Nemeč, B., Kramberger, A., Wörgötter, F., Aksoy, E.E., Papon, J., Haller, S., Piater, J., Krüger, N.: Teaching a robot the semantics of assembly tasks. *IEEE Trans. Syst. Man Cybern. Syst.* **48**(5), 670–692 (2018)
17. Tsai, C., Tsai, Y., Hsu, S., Wu, Y.: Synthetic training of deep CNN for 3D hand gesture identification. In: 2017 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO), pp. 165–170, Prague (2017)
18. Vathos GmbH (2020). <https://www.vathos-robotics.de/>. Accessed 10 Aug 2020
19. Verosim Solutions (2020). <https://www.verosim-solutions.com/>. Accessed 10 Aug 2020

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

