Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 104 (2021) 1257–1262

www.elsevier.com/locate/procedia

54th CIRP Conference on Manufacturing Systems

Synthetic Training Data Generation for Visual Object Identification on Load Carriers

Daniel Schoepflin[a,*], Dirk Holst[a], Martin Gomse[a], Thorsten Schüppstuhl[a]

[a] *Hamburg University of Technology, Institute for Aircraft Production Technology, Denickestr. 17, 21073 Hamburg, Germany*

* Corresponding author. Tel.: +49-40-42878-3012 ; fax: +49-40-4273-14551. *E-mail address:* Daniel.Schoepflin@tuhh.de

Abstract

With visual AI processes relying on individual and context accurate training data, the existing common object datasets and randomization based synthetic data pipelines can only hardly be transferred or applied on specific and narrow industrial tasks. To enable visual AI applications for intralogistics processes, such as supervision or localization of objects, a domain-knowledge driven implementation for generation of context accurate synthetic training data is introduced. With this consideration of process and domain requirements in the data generation pipeline itself, a data-generator for object identification on load carriers is contributed.

Keywords: Production Automation; Intralogistics; Synthetic Training Data; AI Data Generation; Object Identification;

1. Introduction

The lack of training data is widely considered a main challenge for successful implementation of AI applications. For visual tasks, the manual acquisition and labeling process of images may be error-prone and is considered costly [9, 5]. Synthetic training data, e.g. in the form of rendered images, alleviates these problems and may enable broad usability of AI applications. Successful training of AI applications with synthetic data was achieved in various fields such as identification of household objects [15], picking [7] or autonomous driving [16]. Specific applications like assembly supervision [12] or bin-picking tasks have shown principle applicability in industrial environments. Most of the created datasets are publicly available, but not necessarily transferable to similar tasks at a different industrial user. Common object applications, e.g. household object detection, can be applied to multiple different households, as they will share most of the occurring objects. Industrial environments however differ heavily from another. This results in specific and individualistic tasks that require equally individualistic training data. Thus, developers often need to create their own datasets for their specific applications. With synthetic dataset creation tools, variations of object scenes can be created and rendered to obtain image datasets. Due to their aim of universality, the user is tasked with modeling the environment and definition of the variations. Such modeling and recreation are critical for successful training of an application, but again require domain experts and their domain-knowledge and again results in an extensive manual effort. By formalizing the domain-knowledge and interweaving it with a toolbox, the need for manual modeling can be reduced.

Following previous work, we are striving to enable synthetic training data generation for the domain of intralogistics, specifically the transport of components on carrier units. The principle applicability of synthetic data for object detection on a load-carrier [21] was demonstrated. This work aims to display a process analysis and to formulate domain-knowledge to derive an implementable domain-representation. An implementation of this formulation in a toolbox is presented, which may be used to generate training data for AI tasks related to object detection of components in carrier units. It is demonstrated how this tool may be used to enable such applications, without prior need for environmental modeling and with little manual effort for the user.

2. Related Work

With manual data acquisition being both time-consuming and expensive, training AI applications with synthetic image data that is rendered out of 3D models has recently gained popularity [15, 4]. Examples of successful training with synthetic

2212-8271 © 2021 The Authors. Published by Elsevier B.V.

10.1016/j.procir.2021.11.211

training data have been demonstrated in applications concerning autonomous platforms [10, 16] and urban scenes [14], picking objects [25], robotic surgery [3], or identification of household objects [15, 6] and scenes [13]. In industrial scenarios the grasping of robot picking tasks has been learned based on CAD data [24, 8, 2, 7, 11]. Object recognition and assembly tasks with pegs [26] or screws [27, 12] were also successfully trained with synthetic data. In [21] we demonstrated successful training of an object identification AI with synthetic data in intralogistic settings. A methodology for combining digital twins of industrial processes with synthetic data generation was introduced by [1].

Most commonly, the process of synthetic image data generation resolves around a 3D scene that is created in a virtual environment and then rendered to generate the image output. As this output should reflect different scene variations to successfully train the application, certain parameters are varied during the creation process. Typical parameter variations include lighting, camera position, and object positions. Data Synthezising toolboxes as NDDS[1] for Unreal Engine, SynthDet[2] for Unity and similar for Blender[3] provide this functionality of randomizing the image output.

Complete randomization of image data may be useful as shown by [25] and alleviates the need for strict environmental modeling. However, contextual accuracy of the created data may benefit the transfer of the trained AI towards the real application domain [17]. To achieve contextual accuracy, [19] introduce a principle of Structured Domain Randomization, in which scenes are created according to problem-specific probability distributions. Modifying scene creation to match realistic contexts was achieved by [17] by introducing a learning-based scene creation. As they pointed out, having a proper scene grammar leading to contextual accurate scenes benefits the transfer from the synthetic training domain towards the real application domain. As shown in [12], deriving such a grammar requires domain knowledge and a thorough understanding of the variations taking place in real scenarios. The common toolboxes for general data generation do not reflect this need for contextual accuracy and domain-knowledge, leading to manual effort for the user. By generalising the approach for synthetic data for intralogistic object identification as reported in [21], we derive an abstracted and formalized representation of domain-knowledge. Fundamental parts of this knowledge are implemented in a toolbox of similar functionalities as the above.

3. Process Analysis and Scene Grammar Derivation

Intending to link domain-knowledge with a data-generation tool, it is first necessary to derive and formalize this knowledge. We start this derivation by introducing a description of the domain and material-flow with carrier units specifically. Afterwards, basic information types are defined, which the synthetic training data aims to enable. Finally, the scenes of interests are parametrized and formalized with semantic composition rules.

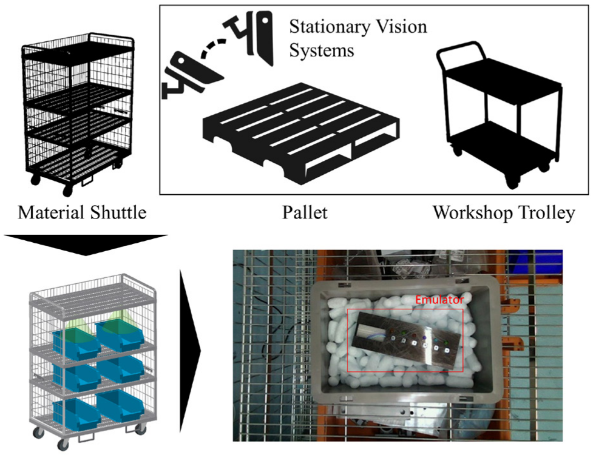

Fig. 1. Examples of the three load carrier types material shuttle (top left), pallet (top mid) and workshop trolley (top right) and visualization of the considered scope for material shuttles (bottom right).

3.1. Carrier Unit Based Material Transportation

Carrier units or load carriers are common on most factory sites and appear in various shapes and types in production supplying intralogistics. Without loss of generality, we base the following analysis on the aircraft final assembly processes [23, 22]. We group the appearing carrier units into four categories, based on their principle appearance and basic functionalities. Figure 1 shows examples for the in this work considered types:

- **Material Shuttle**: multi-level wagons, mostly grid trolleys with variable levels. Sometimes standardized within a company. Often loaded with boxes that are commonly not stacked. Transported with fork-lifts, or as route train.
- **Workshop Trolley**: hand-pushed carts with two levels for short-range transportation of materials.
- **Pallets**: non-rollable base structure of a unit load. Loaded with boxes that are often stacked upon another. Transported with a fork-lift.
- **Special Load Carriers**: often company individualistic carrier units for which no generalized composition grammar can be derived and thus, are considered out of scope for this work.

Goods of medium to large size may be placed directly on units and may be secured with belts or straps. Small and medium-sized goods however are often transported in small load carriers (boxes) that in turn are loaded on these carrier units. These boxes can be filled with various packaging infills like bubble wrap, cartonnage, or flips. Training an AI application to handle object identification tasks despite such challenging situations, will be the main contribution of this work. Other work related transport of such carrier units in intralogistic settings, challenges the task of localizing the units [20]. Here, we ad-

[1] https://github.com/NVIDIA/Dataset_Synthesizer, October 2020
[2] https://github.com/Unity-Technologies/SynthDet, October 2020
[3] https://github.com/921kiyo/3d-dl, October 2020

dress AI based visual tasks that revolve around the identification and localization of transported material. Figure 1 visualizes the considered types of carrier units and possible visual scope for material shuttles. Enabling AI applications based on these tasks requires training data that resembles the mentioned scenarios. Therefore, to derive which aspects and scopes are relevant to display in the training data, we discuss fundamental AI tasks that are based on automated task use-cases around the introduced carrier units.

3.2. Information Type Analysis

In processes with a high degree of manual touches, such as production supplying material flow, errors occur. Typical ones include loading of the wrong components or a wrong number of loaded components. Visual process supervision enables the detection of such errors and requires identification and localisation information. We distinguish between three fundamental information types. Out of these types, AI tasks can be formed:

- **Event Detection**: Change in the supervised scene due to interaction can be translated into predefined events. The most basic detection information carries no classification of the supervised event, but only the fact that it was detected. Classification of an event requires contextual state information such as a description of the planned event. An un-/loading event detection e.g. can be triggered once a camera detects the presence of a tray. Classifying this event further e.g. loading of a certain component requires additional identification information.
- **Identification**: Distinguishing and identifying objects within the image requires classification information. The elementary information indicates whether a certain object is present in the image. Combining this basic information and determining whether multiple objects are present simultaneously quickly allows for contextual information and in turn enables evaluation applications. As such it is possible to determine whether a shipment is complete.
- **Localization**: In the context of visual AI applications, the term object detection is not only used for classification tasks but may also indicate segmentation and drawing of a bounding box around objects. As this is location information, we distinguish it from the identification information types. Strictly defined the information whether an object is within the image (classification) already infers a basic localization, however, our basic localization information is set to be position information. Higher-level information includes orientation information, resulting in a full 6D-pose.

Acquiring these information types through visual AI applications allows not only for automated supervision [12, 21] but also may enable other applications like automated handling, e.g. unloading of components [7]. Another possible application may be the inspection of components with regards to damages and such. This however, is considered out of scope for this work.

3.3. Parametrization of the Scene Variations

Parameters, with which a descriptive representation of the real scenes can be formalised, are necessary to rebuild the scenes in a digital representation. Based on the above described load carrier based material transport, three settings are identified, out of which descriptive parameters can be extracted. Visual indication of those settings is shown in figure 2:

- **Simple Setting**: objects and components are placed on a uniform background such as anti-slip mats. Besides material changes leading to differences in texture and color of the background, parameter variations occur with the placing of the object on such a mat. Objects can occur in different translatory positions as well as rotational placements. However, many components are limited in their contact points to the mat, to which they are forced by their axes of inertia and gravity. This leads to a discrete number of stable resting states, with as little as one axis for rotational degree of freedom.
- **Intermediate Setting**: multiple components are transported in boxes or cartonnages. Those are also subject to the same placement restrictions as the simple cases above. Variant lighting conditions as well as shadowing caused by boxes leads to more challenging scenery depictions.
- **Complex Setting**: some shock-sensitive components may be transported in boxes filled with packaging flips or in bubble wrap. Some components are transported by placing them between struts of a support structure. Additional straps may be used to secure the components. Besides creating complex scenery by adding a complex setting of other objects, they also may allow the components to be placed in different positions than the previous two settings. Additionally, lighting situations are more complex, due to local shadowing. Finally, settings are considered complex, if objects out of context appear, e.g. transport sheets or tools.

This description of scene settings implies parameters to describe and differentiate scenes. For the simple scene settings, we identified four basic parameters, needed to describe the scene: **Position** of each object in the scene. In the simple setting the position of the to be identified objects is restricted by the limitations of the load carrier; **Orientation**: placing objects in the scene, requires defining the orientation of those objects. For simple settings, we assume only one rotational degree of freedom; simple settings contain only simplistic or uniform **backgrounds**; due to the simple scene itself, the applied **lighting** causes little shadows. In these settings, it is considered to be uniform and little directional.

The above parameters are re-used to define intermediate settings but within a more extensive formulation. This caused by the additional parameter **packaging**, which causes restrictions in the positional arguments. Further, the packaging causes casting of shadows which influences the appearance of the scene quite heavily.

Due to more complex packaging settings, the objects may be oriented in highly variational poses caused by the packaging

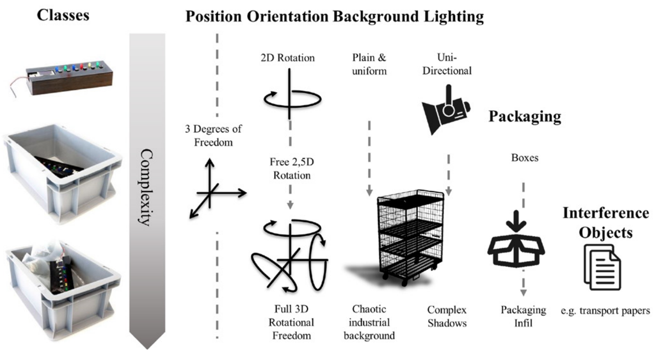

Fig. 2. Differentiation of the analysed scene complexities and the parameters needed to described the respective complexity.

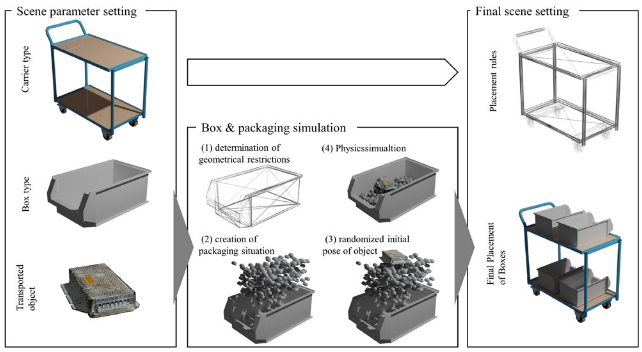

Fig. 3. Flow-Chart visualization of the scene composition rules on the example of a workshop trolley. by choice of the trolley and box model, placement restrictions are defined, as indicated by the wireframe models. To ensure physic accurate placement of components and packaging in the box, a physicssimulation is performed.

scenario, e.g. by packaging-flips. In complex scenarios, highly overloaded backgrounds may appear that bring entire logistics or production environments in the scope of the scene. Also, settings are considered complex due to one additional parameter, the presence of **interference objects** in the scene, papers, or tools that may appear unpredictable in production near environments.

With this set of parameters defined, we are now able to recreate the transport scenes digitally. To automate this scene creation process, composition rules have to be defined that link the parameters to a 3D scene schematic.

3.4. Semantic Composition Rules

In this work the composition of the scene, i.e. specifying the position and orientation for each object, is aimed to be contextual accurate in comparison to fully randomized approaches [25]. Following the precedence set by [17] this requires semantic rules. In this discussed case these rules indicate constraints between a to be placed 3D object and its environment as well as semantic relations between two objects. We define general rules for scenes representing components transported in a box and semantic rules for each of the carrier unit use-cases from subsection 3.1. In the following, we present the abstract formulation of those rules:

Placement regions: Boxes or components are placed on a planar region. Regions are defined by the model of the load carrier. Within one region all components/boxes are placed in a non intersection manner. In figure 3 the chosen trolley contains two placement regions, one on each shelf.

Boxes: If objects are placed in a box, the box opening is oriented upwards. Objects are placed inside the box. Boxes contain placement volumes as indicated in figure 3, within which objects are to be placed. Boxes may be filled with one kind of packaging material.

Packaging flips: if objects are placed in a bed of packaging flips, the packaging flips are to be filled into the transportation box, before the main objects are placed in the box. Packaging flips fill the box between 10 and 50 %.

Anti Slip Mats: if objects are placed on anti-slip mats, the objects are to be placed on top of the mats

Bubble Wrap: if objects are to be placed in bubble wrap, the pose of the object is affected by the underlying material.

Object Poses: the poses of objects are to follow physical accuracy. A physics simulation is to be performed. Objects are not allowed to intersect with other objects. Scenes contain only objects at rest. Objects may contain preferred orientations caused by distinctive main axis of inertia. Objects may not be placed in unstable points of rest. Objects may contain orientation restrictions, e.g. components with displays or sensitive areas.

3.5. Viewing Properties and Visual Scope

Once a 3D scene is created, visual representations have to be rendered. For this, viewpoint and direction parameters have to be set. As this work pursues the generation of synthetic data for tasks resolved around transported materials, detection of transport carriers itself is considered out of scope. We, therefore, assume a visual scope focused around or in load carrier as seen in figure 1. In order to fulfill the principle supervision tasks, variations of these situations are limited to quantitative changes of view parameters. We describe the camera and light objects with spherical coordinates, with the center located in the transport setting. For each light and camera object, the parameters polar angle, azimuth, and radius are defined. Further for the camera, resolution and aspect ratio have to be specified.

4. Implementation

The derived formalized description of scenes can now be implemented in a toolbox that then can be used to create training data for the presented applications. We chose the open-source 3D creation suite Blender for implementation. The toolbox aims to automate as much as possible with minimal manual input. This is achieved by utilizing the derived composition rule but results in limitations with respect to the principle scene variability compared to other tools. At this stage, the toolbox contains an implementation for the essential box simulation from figure 3 on a material shuttle. Future releases will include variations

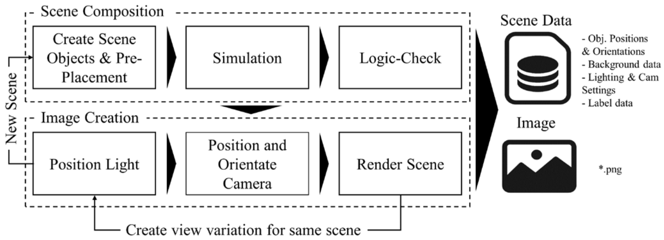

Fig. 4. Flow-Chart representation of the toolbox. After specification of the parameters, the scene composition with physics simulation is performed. Afterwards variants of that scene are rendered with different light and camera settings. Besides the image, all scene composition parameters are stored in a database.

on different load-carriers. We present the functionality of the developed toolbox with consideration to the transport setting as described in figure 3. The toolbox and user documentation can be found online [4].

After a user defines quantifications for the parameters of the scene, the toolbox iterates through a generation sequence as shown in figure 4: Objects are loaded as *.obj* file in the scene and scaled with to a uniform scale to avoid unit-mismatches. Afterwards, the objects are placed according to the grammar rules from 3.4. The packaging situation is recreated with a rigid body physics-simulation. To verify contextual correctness of the simulated scene, a Logic-Check is implemented that evaluates the created scene with respect to relative position of objects and box as well as field-of-view of the camera. If objects appear inconsistent with the scene grammar and the users' parameters, the created scene is discarded. E.g. images with objects out of the box, not visible objects, or similar occurrences are deleted.

Once the scene is created, the image is rendered. Another loop is introduced, that varies the light position, camera placement, and orientation according to the visual scope as defined by the user. Afterwards, the rendering is performed. Our toolbox supports the Blender Ray-Tracing Render-Engine *Cycles* as well as the Real-Time Render-Engine *Eevee*. Created images are exported as *.png* and the label information is stored in a database. Alongside the primary label, e.g. class and bounding box, every scene creation parameter is saved in the database.

5. Demonstration

With the presented toolbox a dataset is generated to enable an object identification task of two different components on a Material Shuttle used in a production site. As user, the *.obj* file-paths of those objects and name the packaging variations are specified. Packaging scenarios with flips, bubble wrap, and no packaging are chosen. From the production site statement, the definition of the lighting setting is inferred. The backgrounds are taken from the background repository for material shuttles. The object 3D data is captured with a Artec 3D Scanner. Cycles Rendering Engine is chosen for photo-realistic rendering.

For each class 21 scene variations are composed by the toolbox. Considering the camera and lighting variations in sum 740 images per class are rendered. Examples of the synthetic created images are shown in figure 5.

A VGG-16 network initialized with the Image-Net trained weights is chosen. A classificator of a 1024 node Dense layer, 20% Dropout and a single node classification output is constructed. The top classificator is trained for five epochs with a batch size of 32 and the adam optimizer. Afterwards, the entire network is trained with the same batch-size and an SGD optimizer for ten epochs. Unlike similar work [21, 12], no real fine-tune data is needed to address the lack of domain adaption methods.

The network is tested against a test-set containing 130 real images of object 1 and 164 images of object 2. The objects are placed in various box shapes and different packaging materials. Additionally, lighting situations were created. Camera positioning is done similar to figure 1. Examples of the test sets are shown in figure 5. The test-set contains scenes that are similar to the re-enacted scene but differ in key aspects such as cartonage packaging instead of transportation boxes, scenes without any boxes and cartonages and wrong objects in the field of view.

The trained neural network is evaluated on the test-set, with a resulting classification accuracy of 99%. A single image out of the 294 test-images was predicted with the false class. Due to the contextual similarity between the synthetic and the application domain, a successful transfer without additional domain adaption techniques was achieved. Therefore, the derived domain-knowledge based approach for contextual data generation is considered successful.

Future work with the toolbox should reflect the variability of the enabled AI applications as discussed in subsection 3.2 and is to be evaluated against a broader range of scene variations.

Fig. 5. (left) examples of the created synthetic training data; (right) examples of the real world test data. The test-set contains scenarios that were not re-enacted in the synthetic data approach (e.g. certain type of boxes). Successful handling of such scenes infers validity of the synthetic training data for use in this domain.

6. Conclusion and Outlook

In this work, we presented a formalization of domain-knowledge for the transport of components with load carriers.

[4] https://github.com/TUHH-IFPT/Toolbox-Intralogistic-AI-Data

This formalization is used to derive a toolbox for synthetic data creation in the application domain of load carriers, with which visual identification tasks can be trained. We demonstrated the applicability to train an object identifier for a material shuttle based transportation scenario.

Data generated from the presented toolbox may be utilized to enable digitalized intralogistic processes such as integration with a smart load carrier [22] or conveyor belts [18]. Future work will focus on expanding the functionality of the toolbox towards multiple use-cases and different environments.

CRediT author statement

Daniel Schoepflin: Conceptualization, Methodology, Software, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization, **Dirk Holst:** Conceptualization, Methodology, Software, Writing - Review & Editing **Martin Gomse:** Writing - Review & Editing, Supervision, **Thorsten Schüppstuhl:** Writing - Original Draft, Supervision, Resources, Funding acquisition, Project administration

Acknowledgements

Research was funded by the German Federal Ministry for Economics and Energy under the Program LuFo V-3 DEPOT.

References

[1] Alexopoulos, K., Nikolakis, N., Chryssolouris, G., 2020. Digital twin-driven supervised machine learning for the development of artificial intelligence applications in manufacturing. International Journal of Computer Integrated Manufacturing 33, 429–439.

[2] Brucker, M., Durner, M., Márton, Z.C., Bálint-Benczédi, F., Sundermeyer, M., Triebel, R., 2020. 6DoF Pose Estimation for Industrial Manipulation Based on Synthetic Data, in: Proceedings of the 2018 International Symposium on Experimental Robotics, pp. 675–684.

[3] Cartucho, J., Tukra, S., Li, Y., S. Elson, D., Giannarou, S., 2020. Visionblender: a tool to efficiently generate computer vision datasets for robotic surgery. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization , 1–8.

[4] Dahmen, T., Trampert, P., Boughorbel, F., Sprenger, J., Klusch, M., Fischer, K., Kübel, C., Slusallek, P., 2019. Digital reality: a model-based approach to supervised learning from synthetic data. AI Perspectives 1, 1–12. doi:10.1186/s42467-019-0002-0.

[5] Dekhtiar, J., Durupt, A., Bricogne, M., Eynard, B., Rowson, H., Kiritsis, D., 2018. Deep learning for big data applications in cad and plm – research review, opportunities and case study. Computers in Industry 100, 227–243.

[6] Dwibedi, D., Misra, I., Hebert, M., . Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. URL: http://arxiv.org/pdf/1708.01642v1.

[7] El-Shamouty, M., Kleeberger, K., Lämmle, A., Huber, M., 2019. Simulation-driven machine learning for robotics and automation Simulations. Technisches Messen 86. doi:10.1515/teme-2019-0072.

[8] Fang, K., Bai, Y., Hinterstoisser, S., Savarese, S., Kalakrishnan, M., . Multi-task domain adaptation for deep learning of instance grasping from simulation, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 3516–3523. doi:10.1109/ICRA.2018.8461041.

[9] Gaidon, A., Lopez, A., Perronnin, F., 2018. The Reasonable Effectiveness of Synthetic Visual Data. International Journal of Computer Vision 126, 899–901. doi:10.1007/s11263-018-1108-0.

[10] Gaidon, A., Wang, Q., Cabon, Y., Vig, E., 2016. VirtualWorlds as Proxy for Multi-object Tracking Analysis, in: 29th IEEE Conference on Computer Vision and Pattern Recognition, Piscataway, NJ. pp. 4340–4349. doi:10.1109/CVPR.2016.470.

[11] Georgakis, G., Mousavian, A., Berg, A.C., Kosecka, J., 2017. Synthesizing Training Data for Object Detection in Indoor Scenes, in: Robotics: Science and Systems 2017. URL: http://arxiv.org/pdf/1702.07836v2.

[12] [accepted] Gierecker, J., Schoepflin, D., Schmeedemann, O., Schüppstuhl, T., 2021. Configuration and Enablement of Vision SensorSolutions through a Combined Simulation Based Process Chain, in: Annals of Scientific Society for Assembly, Handling and Industrial Robotics, Springer Vieweg.

[13] Handa, A., Patraucean, V., Badrinarayanan, V., Stent, S., Cipolla, R., . SceneNet: Understanding Real World Indoor Scenes With Synthetic Data, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/CVPR.2016.442.

[14] Hattori, H., Lee, N., Boddeti, V.N., Beainy, F., Kitani, K.M., Kanade, T., 2018. Synthesizing a Scene-Specific Pedestrian Detector and Pose Estimator for Static Video Surveillance. International Journal of Computer Vision 126, 1027–1044. doi:10.1007/s11263-018-1077-3.

[15] Hinterstoisser, S., Pauly, O., Heibel, H., Marek, M., Bokeloh, M., . An Annotation Saved is an Annotation Earned: Using Fully Synthetic Training for Object Instance Detection, in: 2019 IEEE International Conference on Computer Vision Workshop (ICCVW).

[16] Johnson-Roberson, M., Barto, C., Mehta, R., Sridhar, S.N., Rosaen, K., Vasudevan, R., . Driving in the Matrix: Can Virtual Worlds Replace Human-Generated Annotations for Real World Tasks? URL: http://arxiv.org/pdf/1610.01983v2.

[17] Kar, A., Prakash, A., Liu, M.Y., Cameracci, E., Yuan, J., Rusiniak, M., Acuna, D., Torralba, A., Fidler, S., 2019. Meta-sim: Learning to generate synthetic datasets, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 4550–4559. doi:10.1109/ICCV.2019.00465.

[18] Mörth, O., Emmanouilidis, C., Hafner, N., Schadler, M., 2020. Cyber-physical systems for performance monitoring in production intralogistics. Computers and Industrial Engineering 142, 106333.

[19] Prakash, A., Boochoon, S., Brophy, M., Acuna, D., Cameracci, E., State, G., Shapira, O., Birchfield, S., 2019. Structured Domain Randomization: Bridging the Reality Gap by Context-Aware Synthetic Data, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, [Piscataway, NJ]. pp. 7249–7255. doi:10.1109/ICRA.2019.8794443.

[20] Schoepflin, D., Gomse, M., Schüppstuhl, T., 2020. Towards Visual Referencing for Location Based Services in Industrial Settings, in: Proceedings of the 52nd International Symposium of Robotics, VDE.

[21] [accepted] Schoepflin, D., Iyer, K., Gomse, M., Schüppstuhl, T., 2021a. Towards Synthetic AI Training Data for Object Identifiers in Intralogistic Settings, in: Annals of Scientific Society for Assembly, Handling and Industrial Robotics, Springer Vieweg.

[22] [accepted] Schoepflin, D., Julian, K., Gomse, M., Schüppstuhl, T., 2021b. Smart Material Delivery Unit for the Production Supplying Logistics of Aircraft, in: Procedia Manufacturing, Elsevier.

[23] Sliwinski, M., Raabe, C.M., Braun, P., Hinckeldeyn, J., Kreutzfeldt, J., 2020. Modulare Ladungsträger für den Kleinteiletransport. ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb 115, 418–421.

[24] Thomas, G., Chien, M., Tamar, A., Ojea, J.A., Abbeel, P., . Learning Robotic Assembly from CAD, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 3524–3531. doi:10.1109/ICRA.2018.8460696.

[25] Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P., 2017. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). doi:10.1109/IROS.2017.8202133.

[26] Triyonoputro, J.C., Wan, W., Harada, K., . Quickly Inserting Pegs into Uncertain Holes using Multi-view Images and Deep Network Trained on Synthetic Data, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5792–5799. doi:10.1109/IROS40897.2019.8968072.

[27] Židek, K., Lazorík, P., Piteľ, J., Hošovský, A., 2019. An Automated Training of Deep Learning Networks by 3D Virtual Models for Object Recognition. Symmetry 11, 496. doi:10.3390/sym11040496.