

Generating Physically Sound Training Data for Image Recognition of Additively Manufactured Parts

Tobias Nickchen
Paderborn University

t.nickchen@sicp.uni-paderborn.de

Stefan Heindorf
Paderborn University

heindorf@uni-paderborn.de

Gregor Engels
Paderborn University

engels@uni-paderborn.de

Abstract

In recent years, Additive Manufacturing (AM) has evolved from a niche technology for prototyping to a well-known industrial production process. In this work, we focus on Selective Laser Sintering (SLS)—one of the leading AM techniques. While SLS has many advantages, the simultaneous manufacturing of multiple components requires the subsequent recognition of components which must be done manually in today's production processes. While approaches for automatic, sensor-based object recognition have been proposed, e.g., based on Convolutional Neural Networks (CNNs), they assume the availability of real-world photos which is not given in the setting of Additive Manufacturing. Hence, we develop an approach to render realistic virtual images and demonstrate their suitability to recognize real-world objects. Although often done in the machine learning community, orienting the objects randomly generates many orientations that are physically impossible and cause distracting noise in the training process. Hence, we pay particular attention to generate physically sound training data and we demonstrate that our approach significantly improves the recognition rate compared to traditional approaches.

1. Introduction

Additive Manufacturing (AM) is a growing manufacturing process with great developments in recent years. For example, powerful AM machines and new interfaces for automation and series production have made it significantly easier to integrate AM into complex process chains leading to growing production volumes for AM service providers. However, there are still some challenges with integrating AM steps into fully automatic process chains—particularly with powder-bed-based Selective Laser Sintering (SLS) steps [16]. A special characteristic of SLS is that components are manufactured in batch mode, i.e., many components are manufactured simultaneously three-dimensionally nested in the build chamber, requiring to identify and sepa-

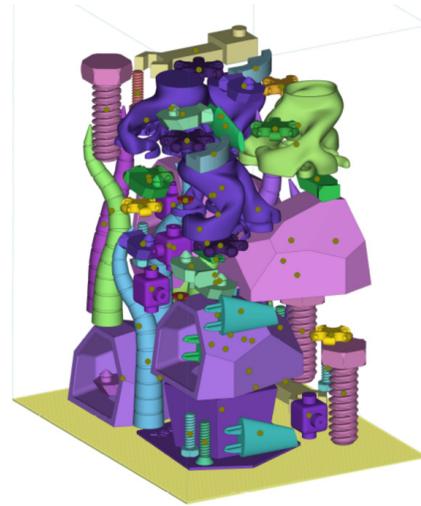


Figure 1. Selective Laser Sintering (SLS) build job consisting of several different CAD models.

rate the different components subsequently, e.g., to forward them to different milling, polishing or painting processes.

Identifying and separating individual components is most commonly done manually as of today. In this paper, we propose an approach to automate this step with Computer Vision (CV) systems. While CV systems are already used to automate processes, CV systems are often manually optimized for, e.g., the recognition of a specific component and cannot identify arbitrary components produced in an AM step. To circumvent this limitation, we resort to machine learning-based 3D object recognition and improve its recognition performance in the setting of AM by improving its training data. Existing approaches for image-based object recognition are commonly trained with random orientations of components neglecting that many orientations would never occur in practice. For example, a pen would never stand on its tip. In this paper, we generate physically sound training data which only includes physically possible orientations and demonstrate that our approach significantly outperforms approaches based on random training

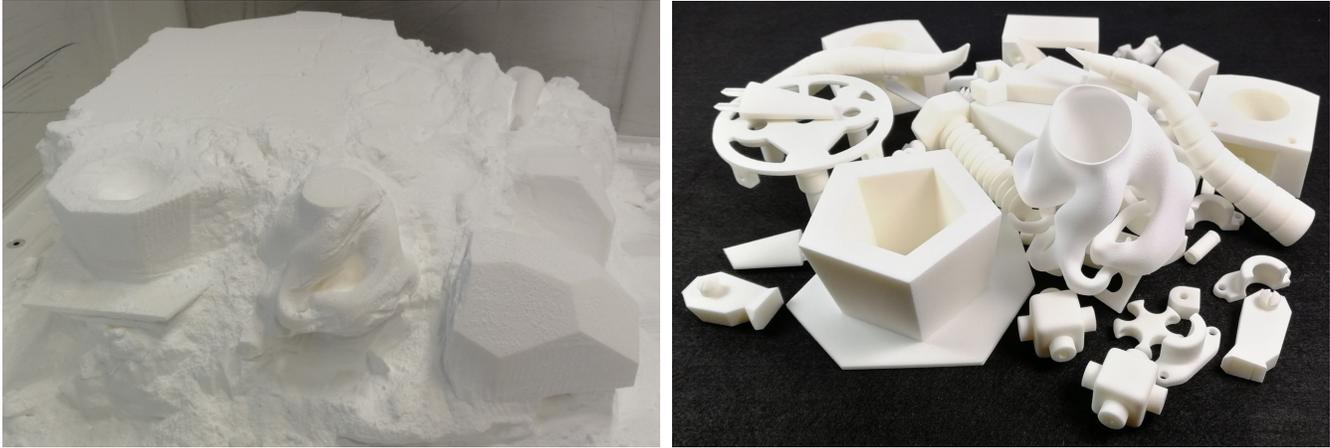


Figure 2. Powder cake with components (left). Clean components after sandblasting (right).

data—with and without common augmentation techniques. Moreover, we demonstrate that our approach is able to recognize objects in real-world images although it was trained on virtually generated images.

In Section 2, we describe the task of component recognition in the setting of AM. Section 3 describes related work. In Section 4, the neural network architecture RotationNet which we use for our experiments is explained. Section 5 describes different approaches of generating training data before Section 6 evaluates and compares the different approaches. In Section 7, we discuss our results.

2. Task: SLS Component Recognition

The necessity to recognize produced components in SLS arises from the production in batch mode, i.e., different objects being produced at the same time in the build chamber of an SLS machine. For the build job preparation, a batch of CAD models is nested in the virtual build chamber for optimal use of the available space (see Fig. 1). SLS is a powder-bed-based AM technique, in which the components are formed layer-wise. The powder is applied in layers of about 0.1 mm and a laser melts the material where components are to be produced. The final components lie together in the so-called powder cake from which they are extracted during powder removal (see Fig. 2 left). After sandblasting, the clean components (see Fig. 2 right) must be further processed individually according to their requirements.

For this purpose, the components must be recognized, i.e., the physical components must be assigned to the corresponding digital 3D models. While this can be done manually for a small number of pieces, for hundreds or thousands of produced parts this becomes impractical and the manual work steps have to be automated, e.g. with a CV recognition system. CV systems are widely used in assembly lines, e.g., for automatic grasping of objects by robots. However, to-

day's systems are often manually optimized for recognition and grasping of a fixed amount of different parts. Due to the daily varying production in AM, this manual optimization is impossible and a data-driven system has to be adopted.

We envision the following structure of a recognition system. The system consists of a separation station and a conveyor belt with an integrated scanning area. After separation, the manufactured objects are transported by the conveyor belt through the scanning area where multiple cameras sense the object from elevated viewpoints (see Fig. 3) and a recognition system assigns the physical objects to the virtual 3D models. The main challenge is to recognize the objects regardless of their orientation. Subsequently, the post-processing steps can be executed.

While a first commercial approach recently emerged for this task (AM Flow [1]), little is known about its internal realization. To the best of our knowledge, it has never been evaluated on a standardized, publicly available data set.

3. Related Work

Related work to our approach can be divided into work on computer vision in industrial applications, machine learning-based object recognition, and data augmentation techniques.

3.1. Industrial Object Recognition

Computer vision systems are nowadays widely used for several different tasks in industrial applications. For example, the systems are used for quality control in the automotive industry [10] and for automatic grasping of objects [3].

Mazetto et al. [10] describe an approach for verifying assemblies in automotive assembly lines with an image-based Deep Learning (DL) architecture to detect different objects like brake disks and calipers. The system is trained on manually labeled images of correct and incorrect assemblies which are taken from the perspective of the assembly control.

Farag et al. [3] present a DL-based method for automatic grasping and localization of 3D objects. The system detects objects placed on a surface using a camera installed above the objects. Similar to Mazetto et al. [10], the system is trained on labeled images. The objects which must be gripped are placed in different positions and photos are taken from the scene. The system learns to detect and localize the objects.

Both approaches are trained on real images. However, real images cannot be used in the setting of AM since the physical objects do not exist at training time yet and we have to rely on virtually generated images which might differ in terms of component orientations, perspectives and lighting.

3.2. Machine Learning-Based Object Recognition

As outlined in Section 2, our goal is to employ approaches from the field of 3D object recognition for our purpose of recognizing additively manufactured parts. The ModelNet leaderboard [12] provides an overview of current state-of-the-art approaches which can be divided into image- and 3D data-based approaches. The approaches are compared using the object recognition task of the ModelNet40 benchmark data set, which consists of 40 different model classes. In this paper, we focus on the image-based approach RotationNet [5], which achieves state-of-the-art performance on the ModelNet40 recognition task. Using RotationNet as a reference approach, we show that the generation of training data has a large effect on the recognition rate and we can significantly improve the recognition rate by generating physically sound training data compared to randomly generated data.

Besides the image-based approach RotationNet, 3D data-based approaches, e.g., based on point clouds [13, 7, 26] or voxel representations [9, 14, 18] might be used for the recognition of 3D models. While these approaches have the potential to yield a higher recognition rate when using a sufficiently high resolution, their exorbitant hardware requirements for sensors and computing power hinder a widespread adoption in practice as of today. Moreover, they suffer from the same fundamental problem that they must be trained on generated representations of components in certain orientations. Hence, even though we focus on image-based approaches, we are optimistic that our generation of physically sound training data would benefit 3D data-based approaches, too.

3.3. Data Augmentation for Object Recognition

Since our training images must be rendered from CAD models, their appearance (slightly) differs from the *real* photos taken after the components' manufacturing and we are faced with a problem of transfer learning [11, 17, 24]. We tackle this problem with data augmentation techniques [2, 19]. Common techniques proposed in the literature include geometric and photometric transformations like flipping, rotation, cropping, translation, noise injection, color space

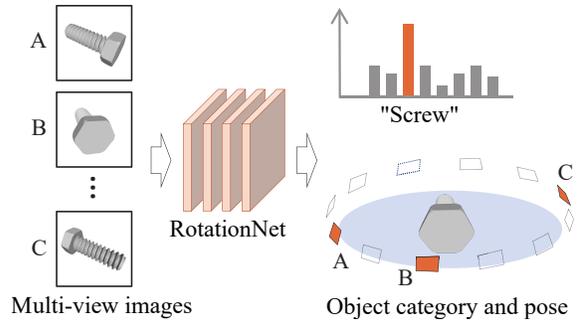


Figure 3. Data representation of RotationNet[5]

transformation, histogram equalization, blurring, brightness changing and sharpening. We employ them as baselines and show that our novel technique of generating physically sound training data significantly outperforms them. Approaches employing domain randomization [22, 23], e.g., placing the objects within different virtual environments with distracting objects and textures, appear less relevant for our purposes since we have a fixed environment with black background and fixed light sources and cameras.

4. RotationNet

In this section, we briefly explain the state-of-the-art approach RotationNet[5] which forms the basis of our evaluation. It is a multi-view-based approach, i.e., multiple images from different perspectives are taken and used as joint training data, thus increasing the amount of available information compared to single-image approaches where parts of an object might be occluded. One of the first approaches in this direction was Multi-View Convolutional Neural Network (MVCNN) for 3D shape recognition [21]. The approach achieved an accuracy of 90.1% on the ModelNet40 data set[25]. Since then, new architectures based on MVCNN have been published. Currently, RotationNet achieves state-of-the-art performance on the ModelNet40 dataset with an accuracy of 97.37%.

The system is trained with a fixed number of viewpoints v_i with $i \in \{1, \dots, K\}$ (see Fig. 3) and for each viewpoint v_i , an image x_i is recorded along with its correct class label y . To increase robustness, the viewpoints are treated as latent variables and the system jointly learns the viewpoints and object labels. For our experiments, we use $K = 12$ viewpoints. Per experiment, we use the same model parameters and the same amount of training instances per object. We fine-tuned the weights of RotationNet based on a model pre-trained on the Imagenet Large Scale Visual Recognition Challenge (ILSVRC) 2012 dataset [15] and used early stopping to avoid overfitting.



Figure 4. Example of 3D models with high geometric similarity.

5. Dataset Construction

For the recognition of additively manufactured components, at the time of training, no physical objects and therefore no real photos of the objects are available. Therefore, artificial photos have to be generated. Then RotationNet is trained on the artificial photos and applied to real photos at testing time (transfer learning). Without additional assumptions, the artificial photos can only be generated from randomly chosen perspectives, which means that many photos are generated from perspectives that cannot occur in a real recognition station due to the geometric properties of the 3D objects. Therefore, we have developed an approach that calculates the physically sound orientations of 3D models and uses only these orientations to generate the virtual images. To the best of our knowledge, there are no other training data generation approaches for object recognition in the AM domain. For evaluation and comparison, we create training data sets from randomly chosen perspectives and use traditional data augmentation techniques. We show that our approach of using physically sound training data significantly outperforms these baselines.

We construct our data sets in four steps: First, we construct six data sets of 3D models of varying difficulty. Second, we generate different instances for each 3D model—based on physically sound orientations and based on random orientations. Third, we render each instance from multiple viewpoints. Finally, we apply traditional image data augmentation techniques.

Section 5.1 describes our selection of 3D models. Sections 5.2 and 5.3 present approaches for generating object orientations—random and physically sound ones. Section 5.4 explains how we render the virtual images and Section 5.5 outlines techniques of image data augmentation. Section 5.6 details how we take photos of the physical objects for evaluation purposes.

5.1. Build Jobs of 3D Models

We construct six virtual SLS build jobs which contain between 10 and 100 different objects from Thingi10K[27], a database containing 10,000 real 3D models from the AM domain. To cover the whole spectrum, we create build jobs of easily discriminable and of hardly discriminable objects: regarding the former, we randomly sample 30, 50, and 100 objects from Thingi10K and regarding the latter, we manually selected 10, 30, and 50 objects from groups of similar

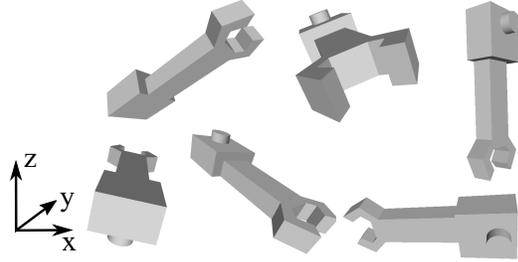


Figure 5. Frontal view of an example object in six random orientations.

objects. Fig. 4 shows an example of hardly discriminable objects. We call our build jobs Random30, Random50, Random100, Similar10, Similar30, Similar50 and we make them publicly available to enable the reproducibility of our results. The two most difficult data sets Similar30 and Similar50 are unlikely to occur in real production but can be used to evaluate the impact of our approach under difficult conditions.¹

5.2. Random Object Orientations

We generate random orientations of objects by calculating uniformly distributed points on a unit sphere. For each point on the sphere, the 3D objects can then be rotated in a way that the vector previously pointing in negative z-direction is aligned to that point [20]. Fig. 5 shows six example orientations of a 3D model from the Thingi10K data set [27] from a frontal viewpoint.

5.3. Physically Sound Object Orientations

The training data can be optimized by aligning the data as closely as possible with the physical data generated in the separation station. As explained in Section 2, the manufactured objects are first separated and then transported to a scanning area by a conveyor belt. When the objects fall onto the conveyor belt, the lying position is influenced by the geometry of the objects. We make use of this fact and determine possible orientations of each object. Along with the camera positions, we can determine possible viewing perspectives for the generation of the virtual training data.

To calculate the possible orientations in which an object is lying on a flat surface, the following steps are performed: (1) calculation of convex hull, (2) checking of object stability, (3) rating based on height of the Center of Mass (CoM).

Convex Hull The convex hull of a 3D object is described by the smallest convex set of points that contains the initial object. The convex hull of the 3D object already shown in the previous section can be seen in Fig. 6. The surfaces of the convex hull form all surfaces which could theoretically tangent a horizontal surface.

¹https://github.com/tobinick-upb/physically_sound_training_data

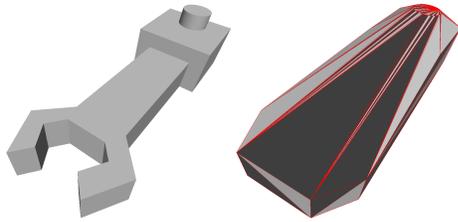


Figure 6. Example object (left) and the corresponding convex hull with its facets (right).

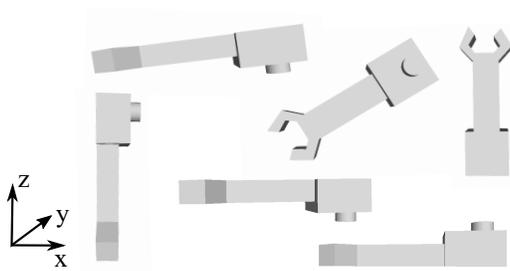


Figure 7. Example object in six different orientations calculated using the convex hull of the object shown from a frontal viewpoint.

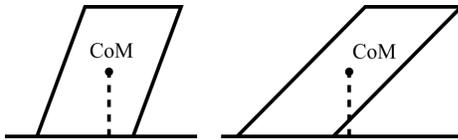


Figure 8. 2D example of a stable (left) and unstable object (right).

This reduces the number of orientations of the 3D objects to be considered. Of basically arbitrary orientations, only those in which the normal vector of one of the surfaces of the convex hull is pointing in negative z-direction, need to be examined. Six example orientations of the 3D model are shown in Fig. 7 from a frontal view. The number of possible orientations can be further reduced in the next steps.

Object Stability The previously calculated set of possible orientations contains all orientations where one of the faces of the convex hull is tangent to a horizontal surface. However, these orientations are not necessarily physically stable. Unstable orientations must be excluded from further consideration. For determining the stability, we employ the objects' Center of Mass (CoM). The CoM is the point of an object at which the distribution of weight is equal in each direction. A 3D object is in a stable position if the projection of the xy-position of its CoM is inside the convex hull of all points which contact the surface. A 2D example of a stable and an unstable object can be seen in Fig. 8.

With that restriction, the number of possible orientations can further be reduced. The effect of the stability check is shown in Fig. 9.

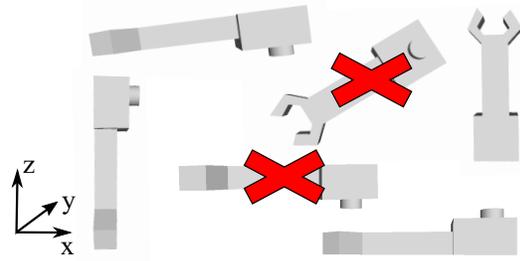


Figure 9. Possible orientations after stability check. The objects are shown from a frontal viewpoint.

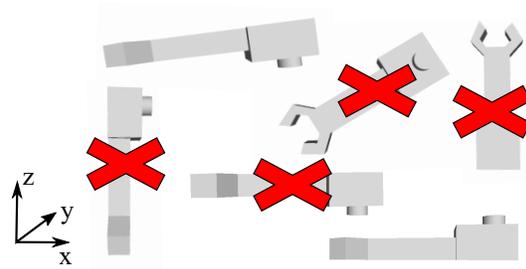


Figure 10. Possible orientations after stability check and exclusion via height criteria. The objects are shown from a frontal viewpoint.

Height of the CoM All orientations of an object calculated in the previous steps are physically possible. Nevertheless, the calculated orientations may contain orientations that will not occur in reality on a conveyor belt. Orientations where a 3D object has a very high center of gravity, compared to other possible orientations, are very unlikely because these orientations have a low stability. Therefore, the objects would tip over on a conveyor belt.

To exclude these unstable orientations, we calculate the height of the CoM of all possible orientations. Orientations whose CoM is four times higher than the lowest one are excluded in our experiments. An example subset can be seen in Fig. 10.

With the help of this procedure, we are able to automatically calculate all possible orientations of a physical object based on the corresponding virtual 3D model. Thereby, we only generate training data from perspectives that can occur in a physical recognition station and thus contribute to the recognition of the respective 3D object.

When training machine learning models, unbalanced training data can lead to a deterioration of the recognition rate and is to be avoided [8]. Hence, regardless of the number of calculated physically sound orientations M , we always generate the same number of training instances N per 3D model, i.e., we create N/M different instances per orientation. Since the physically sound orientations are defined by aligning the normal vector of a plane of the convex hull with the negative z-axis, it is possible to rotate the object around the z-axis without contradicting the calculations for the gen-

eration of physically sound orientations. Since the 12 virtual cameras of RotationNet are positioned in 30-degree steps around the z-axis, we rotate the object in $30/(N/M)$ degree steps. If N/M is not an integer, we round up. In that case, more than N instances are generated. For training, N of these instances are randomly chosen to guarantee that the training data are completely balanced.

5.4. Rendering Images

Based on the calculated orientations, we generate training images which are used as input for the RotationNet model. The rendering of the virtual training images has a big influence on the model’s recognition rate. Therefore, the virtual camera settings, virtual light sources and background have to be aligned to the physical situation as precisely as possible [4]. We render each instance from 12 viewpoints with a rotation around the z-Axis of 30 degrees between two adjacent viewpoints. For the angle to the xy-plane, two different situations have been chosen: (1) an elevation of 45 degrees from the xy-plane and (2) an alternating elevation of 35 and 55 degrees. The second type simulates possibly varying angles in the physical image generation due to inaccuracies in the image generation setting. Su et al. [21] survey several approaches for rendering images from 3D models: Phong Shading, Depth Rendering or Silhouette Rendering. We employ Phong Shading to generate photo-realistic images with details like reflections. Images are rendered at a resolution of 250 times 250 pixels and a perspective camera. We have chosen a black background to maximize the contrast to the white components produced by SLS. As light sources, we employ three virtual lamps placed in elevated positions around the 3D objects to provide bright illumination without undesired reflections. All 3D models are normalized to guarantee consistent views. An example image is shown in Fig. 11a. As rendering software, we employ Blender [6].

5.5. Image Augmentation

In order to make our training procedure more robust and improve the recognition rate on real-world photos, we experimented with data augmentation techniques [2, 19]. We employ the *photometric* transformations blurring (Fig. 11b), brightness/contrast changes (Fig. 11c), histogram equalization (Fig. 11d), noise injection (Fig. 11e) and combinations thereof (Fig. 11f). These transformations are well suited to simulate photometric effects occurring when taking the photos of the physical images. *Geometric* transformations appeared less meaningful: RotationNet already considers different geometric perspectives and our sorting station guarantees that the objects are centered in front of the camera. Thus, based on the basic randomly and physically sound generated data sets, we create six additional data sets for comparison (horizontal flipping, blurring, random brightness contrast (RBC), contrast limited adaptive histogram equal-

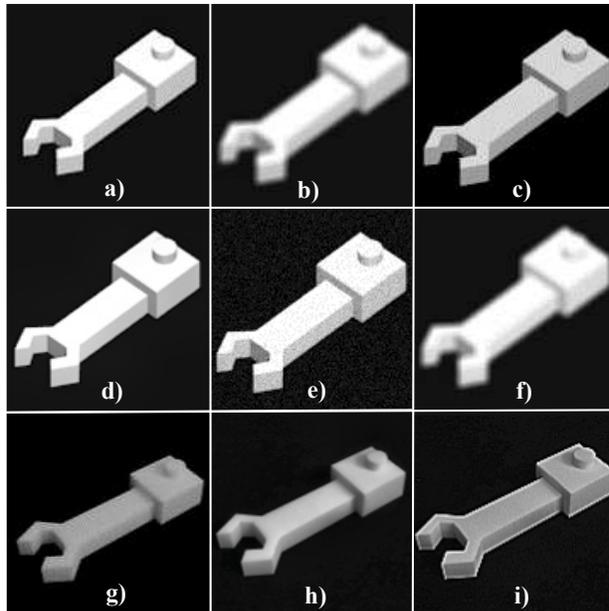


Figure 11. Example of training image raw (a) and with different augmentations (b) to (f) and physical photo raw (g), sharpened (h) and sharpened and histogram-equalized (i).

ization (CLAHE), Gaussian noise, composition of transformations) using the image augmentation implementation provided by Buslaev et al. [2].

5.6. Physical Image Generation

Although we need to render the images virtually before the objects are manufactured, in the end, the recognition rate on real photos counts for our AM application. Hence, we physically produced all objects from two build jobs (our simplest build job Random30 and our most difficult build job Similar50) and took photos of the objects: each object has been placed in its most likely orientations (between 2 and 5 depending on the object’s geometry) and 12 images have been taken per orientation in 30 degree steps around the z-Axis in an elevation of about 45 degrees from the xy-plane. The images have been generated using a hand-held smartphone camera with slight variations of the angles. An example image is shown without post-processing in Fig. 11g, with a sharpening filter in Fig. 11h and with both sharpening filter and histogram equalization filter in Fig. 11i.

6. Evaluation

For evaluating our physically sound training data, we employ the build jobs, object orientations, viewpoints, image augmentation techniques, photos and the model RotationNet as described above. Moreover, to explore how the number of *training* instances affects the recognition rate, we generate two datasets per build job and augmentation technique: one with 60 training instances per object (i.e., orientations) and

Table 1. Random vs. physically sound training data with different augmentations. The trained models were evaluated on photos of physical objects of the Random30 and Similar50 data sets.

	Angle	Inst.	Random30		Similar50	
			Rand.	Phys. S.	Rand.	Phys. S.
Raw	fixed	60	77.61%	98.51%	37.59%	75.94%
RBC	fixed	60	91.04%	97.01%	61.65%	80.45%
Comp.	fixed	60	94.03%	95.52%	62.41%	81.95%
Raw	altern.	60	89.55%	98.51%	53.38%	71.43%
RBC	altern.	60	91.04%	97.01%	66.17%	74.44%
Comp.	altern.	60	95.52%	98.51%	69.92%	79.70%
Raw	fixed	240	88.06%	95.52%	47.39%	66.17%
RBC	fixed	240	97.01%	98.51%	63.91%	81.95%
Comp.	fixed	240	97.01%	98.51%	69.17%	80.45%
Raw	altern.	240	82.09%	98.51%	50.38%	67.67%
RBC	altern.	240	97.01%	98.51%	69.92%	76.70%
Comp.	altern.	240	97.01%	98.51%	69.17%	82.71%
Mix		240	97.01%	98.51%	71.43%	84.21%

one with 240 training instances per object. For the former, we create datasets of 60 training and 20 disjunct validation instances per object and render each of them from 12 viewpoints, yielding 720 training and 240 validation images per object. That means, for our build job Random30, we employ 21,600 training and 7,200 validation images in total. For the latter, we create datasets with 240 training and 30 disjunct validation instances. The *validation sets* are used for early stopping. The *accuracy* is determined per instance since RotationNet calculates the most likely label based on the combination of the 12 views. To account for different numbers of instances per object in our *physical test set*, in which different objects can have different numbers of orientations (unlike in the validation set), the instances are weighted accordingly, i.e., an instance of an object with n orientations is weighted as $1/n$. To enable the reproducibility of our results, the source code of our experiments and the corresponding data is made publicly available.² We provide the six data sets with the original 3D models, scripts for calculating orientations, scripts for rendering of virtual training images and augmentation, scripts for splitting the datasets into train, validation and test sets as well as the RotationNet settings for the optimization solver, network architecture and training/testing commands.

6.1. Evaluation on Real Images

The recognition rate on physical objects from the build jobs Random30 and Similar50 are shown in Table 1. In both cases, we trained the RotationNet model on the raw images without augmentation (raw) and with the augmentations random brightness control (RBC) and composition (Comp.)

²https://github.com/tobinick-upb/physically_sound_training_data

where composition randomly combines two augmentation techniques: one of Gaussian noise or RBC (probability of 50% each) and one of blurring or contrast limited adaptive histogram equalization (CLAHE, probability of 50% each). Besides distinguishing variants with 60 and 240 training instances, we distinguish instances with fixed and alternating vertical camera angles: the former uses a fixed vertical angle of 45 degree from the xy-plane and the latter an alternating angle of 35 or 55 degrees (*altern.* in Table 1). Additionally, a mixed set of best-performing augmentation techniques has been evaluated using 1/4 raw images with fixed angles, 1/4 augmented with RBC and fixed angles, 1/4 augmented with composition and fixed angles as well as 1/4 with composition and alternating angles. For brevity, we do not report the results for CLAHE, Gaussian noise and blurring since they reached lower accuracy than other augmentations techniques. For RBC, CLAHE and Gaussian noise, we employed the standard parameters from the image augmentation library [2]. For blurring, we used a kernel size of 5×5 .

As can be seen in the table, physically sound training data outperforms random training data regardless of the augmentation technique and the number of training samples. On the simple dataset Random30, a high recognition rate can be achieved using image augmentation even without physically sound training data. Nevertheless, physically sound training data still achieve a slightly higher accuracy. For the difficult dataset, the accuracy of the models trained on physically sound training data is significantly higher than the accuracy using random training data. Combining physically sound training data with image augmentation leads to an increase of accuracy from 75.94% to 84.21%. For the randomly generated training data, the effect of augmentation is even higher: the accuracy increases from 37.59% to 71.43%. Overall, the mixed variant leads to the best recognition rate by simulating variations of the physical conditions and avoiding overfitting to the virtual data.

6.2. Evaluation on Rendered Images

In addition to the *realistic* experiment from above, we perform another *artificial* experiment with further datasets (which we could not manufacture due to cost constraints): we train the models on objects from six build jobs as above and *evaluate* them on virtual images: Per object, the validation set consists of 20 additional instances from 12 viewpoints.

Table 2 shows the test results of the models based on randomly (with and without image augmentation) and physically sound training data on the physically sound generated test set for all six datasets. For brevity, we show only the best result in the column data augmentation. Again, the results show that the usage of physically sound generated training data significantly improves the recognition rate of the models compared to randomly generated training data. The models trained on physically sound training data achieve

Table 2. Random vs. physically sound training data. The approach RotationNet was trained on random vs. physically sound training data and evaluated on virtually generated physically sound test data.

	Random	Random w. Augm.	Phys. Sound
Random30	99.79%	100.00%	100.00%
Random50	99.26%	99.00%	100.00%
Random100	92.29%	96.80%	99.82%
Similar10	57.37%	64.38%	100.00%
Similar30	74.64%	83.33%	100.00%
Similar50	63.00%	71.00%	99.25%

over 99% accuracy on all data sets. For the models trained on randomly generated training data, a clear decrease of accuracy can be seen for the data sets with geometrically similar objects. An interesting point is also that the data set Similar10 seems to be more difficult than Similar30 although it includes less objects. Although the image augmentation techniques can again improve the recognition rate, the models trained on physically sound training data outperform the models trained on randomly generated data.

These additional results support the results of the experiments with physical photos and show that especially with increasing complexity of the data, the positive effect of using physically sound training data increases strongly.

These results show that our approach can significantly increase the recognition rate and that our approach can be an important part for the creation of a recognition station of additively manufactured parts. Nevertheless, we found many factors to influence the recognition rate, from the generation of virtual data over training procedures to the recognition using physically generated images. For datasets with many different 3D models and high geometric similarities, all parts of this process have to be optimized.

7. Discussion

In our prototype, we generated the images with a hand-held camera without a fixed recognition station. Hence, the physical training data varies slightly and is not perfectly aligned to the training data. We believe the results could be further improved with a fixed recognition station. Generally, the virtual rendering scene should resemble the physical setting as closely as possible, e.g., in terms of focal length, focal ratio, illumination and positioning of components.

8. Conclusion

Our research shows that the optimization of the training data has a major influence on the classification rate of a component recognition system for additively manufactured parts. Aligning the training data with the physical sensor data enables us to reach recognition rates closer to 100% compared to traditional, random training data generation without fur-

ther assumptions. The positive influence of the physically sound data generation is growing with the complexity of the build jobs and especially with the geometric similarity of the corresponding 3D models. Our work contributes to the full automation of the process step of component recognition to enable fully automatic processes without manual steps.

Generating physically sound training data might not only be important for the recognition of additively manufactured parts. Due to the increasing digitization in industrial processes, it becomes increasingly important to link virtual and physical data in learning-based approaches. Besides the detection of AM components, our approach might be used in industrial assembly lines for applications such as automatic gripping, visual quality control or automated sorting.

In future work, we will compare our 2D multi-view approach to 3D-based approaches, e.g., based on point clouds and voxels, with respect to practical feasibility and the recognition rate of additively manufactured parts.

Acknowledgement

The authors gratefully acknowledge the funding of this project by computing time provided by the Paderborn Center for Parallel Computing (PC²). We would also like to thank the Phoenix Contact Foundation for their great support of our research.

References

- [1] AM-Vision: 3D-part Recognition. Available at <https://am-flow.com/vision/>, Last accessed on 2020-11-03.
- [2] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. Alumentations: Fast and Flexible Image Augmentations. *Information*, 11(2), 2020.
- [3] Mohannad Farag, Abdul Nasir Abd Ghafar, and Mohammed Hayyan ALSIBAI. Real-Time Robotic Grasping and Localization Using Deep Learning-based Object Detection Technique. In *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, pages 139–144. IEEE, 2019.
- [4] Tomáš Hodaň, Vibhav Vineet, Ran Gal, Emanuel Shalev, Jon Hanzelka, Treb Connell, Pedro Urbina, Sudipta N Sinha, and Brian Guenter. Photorealistic Image Synthesis for Object Instance Detection. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 66–70. IEEE, 2019.
- [5] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. Rotationnet: Joint Object Categorization and Pose Estimation using Multiviews from Unsupervised Viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5010–5019, 2018.

- [6] Tang Lee. *Blender Phong: Multi-view Phong Shading*, 2017. Available at <https://github.com/WeiTang114/BlenderPhong>, last accessed on 2020-11-03.
- [7] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-Shape Convolutional Neural Network for Point Cloud Analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019.
- [8] Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and Cristiano Malossi. Bagan: Data Augmentation with Balancing GAN. *arXiv preprint arXiv:1803.09655*, 2018.
- [9] Daniel Maturana and Sebastian Scherer. Voxnet: A 3D Convolutional Neural Network for Real-time Object Recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015.
- [10] Muriel Mazzetto, Luiz FP Southier, Marcelo Teixeira, and Dalcimar Casanova. Automatic Classification of Multiple Objects in Automotive Assembly Line. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 363–369. IEEE, 2019.
- [11] Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.
- [12] Comparison of state-of-the-art 3D ML Algorithms. Available at <http://modelnet.cs.princeton.edu/>, last accessed on 2020-11-03.
- [13] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.
- [14] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning Deep 3D Representations at High Resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3577–3586, 2017.
- [15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet Large Scale Visual Recognition Challenge. *International journal of computer vision*, 115(3):211–252, 2015.
- [16] Manfred Schmid. *Laser Sintering with Plastics: Technology, Processes, and Materials*. Carl Hanser Verlag GmbH Co KG, 2018.
- [17] Ling Shao, Fan Zhu, and Xuelong Li. Transfer Learning for Visual Categorization: A Survey. *IEEE Trans. Neural Networks Learn. Syst.*, 26(5):1019–1034, 2015.
- [18] Tianjia Shao, Yin Yang, Yanlin Weng, Qiming Hou, and Kun Zhou. H-CNN: Spatial Hashing based CNN for 3D Shape Analysis. *IEEE transactions on visualization and computer graphics*, 2018.
- [19] Connor Shorten and Taghi M Khoshgoftaar. A Survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1):60, 2019.
- [20] Cory Simon. *Generating Uniformly Distributed Numbers on a Sphere*. Available at <http://corysimon.github.io/articles/uniformdistn-on-sphere/>, last accessed on 2020-11-03.
- [21] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-View Convolutional Neural Networks for 3D Shape Recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.
- [22] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IROS*, pages 23–30. IEEE, 2017.
- [23] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects. *arXiv preprint arXiv:1809.10790*, 2018.
- [24] Karl R. Weiss, Taghi M. Khoshgoftaar, and Dingding Wang. A Survey of Transfer Learning. *J. Big Data*, 3:9, 2016.
- [25] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D Shapenets: A Deep Representation for Volumetric Shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.
- [26] Kuangen Zhang, Ming Hao, Jing Wang, Clarence W de Silva, and Chenglong Fu. Linked Dynamic Graph CNN: Learning on Point Cloud via Linking Hierarchical Features. *arXiv preprint arXiv:1904.10014*, 2019.
- [27] Qingnan Zhou and Alec Jacobson. Thingi10K: A Dataset of 10,000 3D-Printing Models. *arXiv preprint arXiv:1605.04797*, 2016.