euspen's 21st International Conference & Exhibition, Copenhagen, DK, June 2021
www.euspen.eu

Generation of simulated additively manufactured surface texture data using a progressively growing generative adversarial network

Joe Eastwood[1], Lewis Newton[1], Richard Leach[1] and Samanta Piano[1]

[1] *Manufacturing Metrology Team, Faculty of Engineering, University of Nottingham, UK*

Joe.Eastwood@nottingham.ac.uk

Abstract
In optical surface texture metrology it is often desirable to have access to large quantities of surface data for the purpose of training statistical models. However, these measurements can be time consuming and require user input at many stages of the data acquisition. Generative adversarial networks (GANs) have proven useful in the areas of style transfer and image generation in the field of computer vision. In this paper, we train a GAN on an augmented dataset of additively manufactured (AM) surfaces measured by a focus variation microscope to generate new surface data from a latent input vector. A variety of surface types are included in this dataset to cover a range of expected surface categories generated by a metal AM process. We show, through statistical comparison of areal ISO surface parameters, that the generated surfaces are in fact representative of the real surfaces. These generated surfaces can then be used to generate realistic renderings of AM parts, and for dataset augmentation of machine learning models applied to AM surfaces. While AM surfaces are a useful case study, this approach can be applied to surface data of any type.

surface texture, measurement, metrology, adverserial networks, machine learning, additive manufacturing

1. Introduction

In several fields, including optical metrology, there are many scenarios where it is desirable to have large sets of data, including for the training of statistical models, such as machine learning (ML) networks or to generate realistic surfaces to apply to computer aided design (CAD) models for accurate simulation. An example specific to the application presented in this paper is training a network to detect defects within surface measurements of additively manufactured (AM) parts [1]. A large barrier to obtaining these datasets is simply the logistics of taking such a large set of measurements, typically in the tens of thousands; the manual effort required to generate these datsets renders the endeavour practically infeasible. It would, therefore, be desirable to be able to synthetically generate large datasets at high speed that accurately capture surface textures representative of those seen in reality. The generation of new image data is a highly researched topic in the area of computer vision [2] but there is little work synthesising surface texture data [3-5] and little, if any, that use the ML approaches developed for computer vision tasks.

To this end, we have trained a progressively growing generative adverserial network (PG-GAN) using an augmented dataset of focus variation microscope measurements of AM surfaces to generate data which are new, unique and indistinguishable from an actual measurement. It can be seen in the figures presented later in this paper that there is no discernible qualitative difference between the measurement data and the generated data. Furthermore, through analysis of these data, we show the distribution of areal surface texture parameters created by the PG-GAN is representative of the original data.

1.1. Generative adverserial networks

A generative adverserial network (GAN) is a type of ML model that is often used in the generation of new data, partcularly images [6]. A GAN is composed of two networks: a generator and a discriminator that are trained in a zero-sum optimisation. In simple terms, the discriminator is trained to detect whether a given input image is a real image or a generated image and the generator is trained to generate images that cannot be discriminated from real images. The generator ($G(\mathbf{z})$) takes as input a vector whose elements are taken from a latent manifold of high dimension (typically $\mathbf{z} \in (Z \subset \mathbb{R}^{100})$), a fully convolutional network is then used to upscale and reshape this input into a generated image. The descriminator ($D(\mathbf{i})$) takes an image ($\mathbf{i}$) as input and through a series of convolutional layers, produces a single prediction giving a probability that the input was generated or real. Figure 1 shows a basic GAN archetecture.

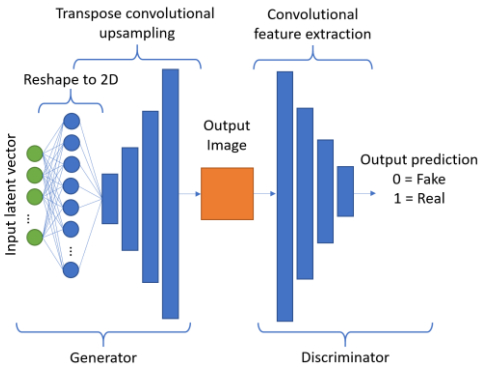


Figure 1. Example GAN architecture. The generator takes a point from the latent space, reshapes and spaitially upsamples this data until the desired output size is achieved. The discriminator takes an input image, downsamples the image through kernel convolution before outputing a prediction (0: fake, 1: real). A batch of images are fed through the network before calculating the loss of the model, which is then used to update the trainable parameters, improving the performance of both the generator and discriminator.

Each block shown in the generator is typically a combination of three layers: a transpose 2D convolutional layer, batch normalisation and a leaky rectified linear unit (ReLU) activation function [7]. Transpose convolution is identical to conventional sliding window kernel convolution; however, each pixel is surrounded by a set of empty pixels determined by a parameter called the stride - this is essentially a method of learned upsampling. The batch normalisation layer normalises pixel values over each batch of inputs. Finally, a leaky ReLU is used as the non-linear activation function. A conventional ReLU function is linear for values greater than zero and zero for all other values. A leaky ReLU instead has a small positive gradient at values less than zero; this allows the network to learn to reintroduce nodes to the model that would otherwise be dropped.

The discriminator is similar to the generator model but contains conventional convolutional layers that spatially downsample the input rather than upsample. The output of this layer is a logistic (sigmoid) function that is trained to predict whether the input image is real (0) or generated (1). We can calculate the loss of the overall model by comparing the predictions given by the discriminator to the ground truth (i.e. was the input actually real or generated?). A sliced Wasserstein distance function [8] is typically used to calculate the loss, which gives the distance between two probability distributions; this can be interpreted as the minimum cost of transforming one distribution into the other. The loss has two components: the 'real loss', how accurately did the discriminator detect real images; and the 'fake loss', how accurately did the discriminator detect generated images? The concatenation of these two losses is used to train the discriminator while the generator is trained just on the inverse of the fake loss - this is essentially how succesful the generator was at fooling the discriminator.

2. Dataset and model summary

For the application presented in this paper, we use an extention of the basic GAN model, referred to as a PG-GAN. This archetecture was first developed by NVIDIA to allow GANs to generate high resolution, photorealistic images [9]. The innovation in this model is to begin by generating low resolution images that are trained against downsampled versions of the training data. As the model converges on a stable output at low resolution, additional layers are smoothly added to the model to produce higher resolution images. This process is repeated until the training data are no longer being downsampled and the full measurement resolution has been achieved. In practice, this smooth adition of new layers to the model is achieved using the architecture shown in Figure 2.

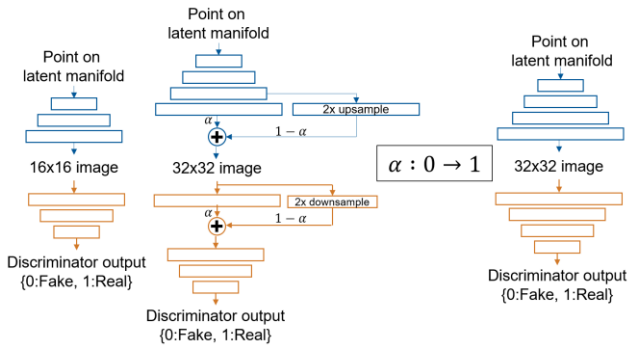


Figure 2. Example showing a PG-GAN growing from simulating 16×16 images to 32×32 images. The parameter α controls how much the new layer contributes to the output; at first α is near-zero, as training progresses the value of α goes to 1, at which point the new layer is fully integrated into the model.

A more detailed explanation of this model can be found elsewhere [9].

The dataset was created by taking a set of surface topography measurements of a ring artefact made from Ti6Al4V using an Arcam A2X electron beam powder bed fusion process. Shown in Figure 3, this part is constructed from a series of planar faces in 10° increments, approximating a cylinder, allowing the dataset to contain surface measurements at a range of measurement angles relative to the build plane, as well as upward and downward facing surfaces.

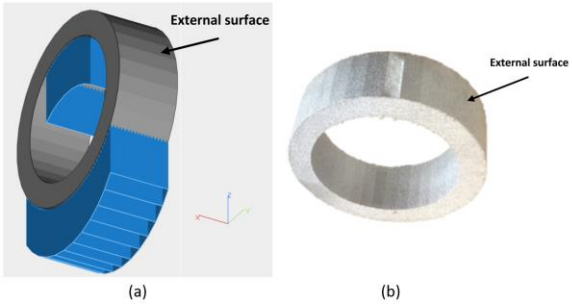


Figure 3. Ring artefact. (a) CAD design, support structures shown in blue. It is expected that the downward surfaces, near the supports, will interact more with the powderbed compared to the upward surfaces. (b) The manufactured artefact.

Fifty-seven measurements of the faces of the ring were taken using a focus variation microscope using the following instrument settings: 20× objective lens (numerical aperture 0.4; field of view (0.81 × 0.81) mm); lateral resolution: 3.51 µm; vertical resolution: 12 nm; ring light illumination; measured area (3 × 3) mm. More details on the generation of these data is given elsewhere [10]. These surface measurements were converted to grayscale images, with the normalised height encoded in the grayscale value at an image size of (1690 × 1693) pixels. These data alone would not be enough to train the model, so the dataset size was augmented by taking (512 × 512) pixel squares at random rotations from random locations on each image. 100 such squares were taken from each image leading to a final dataset size of 5700 images. These data were further augmented during training by permitting the model to reflect the input image about the centre lines of that image.

There are three distinct surface types included in this dataset referred to as top, upward and downward surfaces. Top surfaces are tangent to the build plane and can be indentified by the relatively low spatter and distinctive weld lines. Upward surfaces are the angled surfaces on the top half of the ring where the external surface is in the build direction. In contrast, downward surfaces face in the oposite direction to the build direction resulting in the downward surfaces having greater interaction with the powder bed and thus, a higher concentration of spatter particles, as shown in Figure 4.

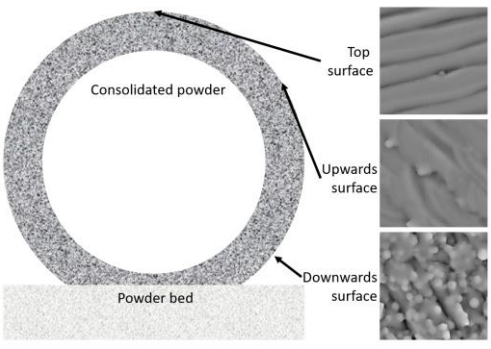


Figure 4. The location and example measurements of top, upward and downward surfaces.

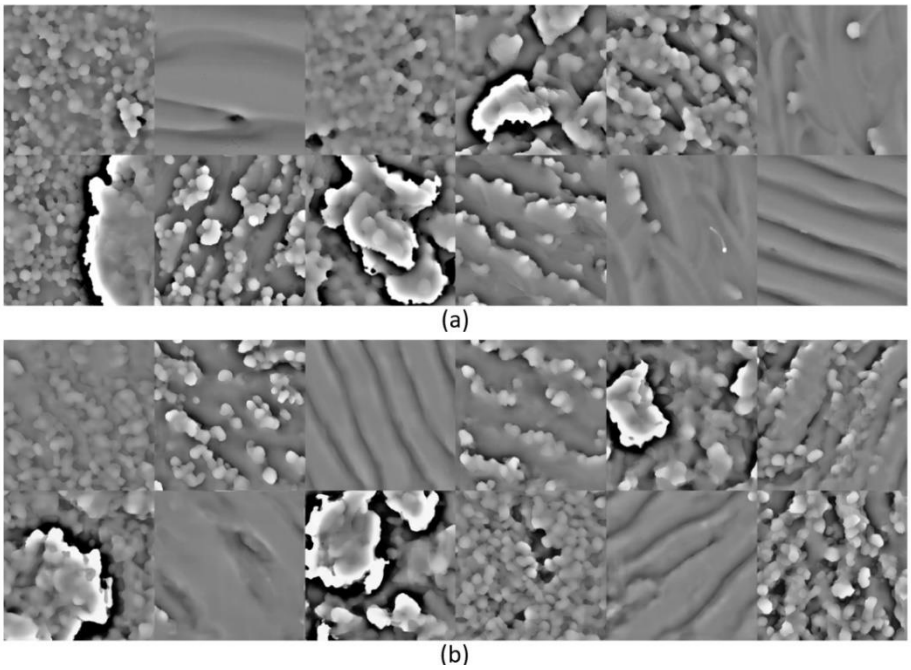


Figure 5. Surface topography measurements with normalised height encoded into image grayscale. (a) Real measurements taken from the training dataset, (b) generated measurements.

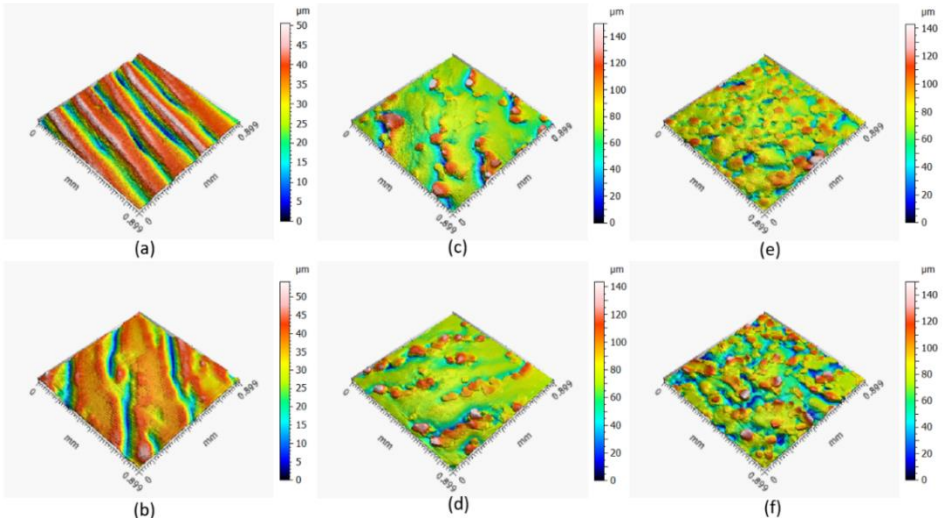


Figure 6. Un-encoded surface topography measurements. (a) Real top surface, (b) generated top surface, (c) real upwards surface, (d) generated upwards surface, (e) real downwards surface, (f) generated downwards image

The model was trained on the Augusta [11] high performance cluster (HPC) on a graphics node with 2 × 20 core processors (Intel Xeon Gold 6138 20C 2.0 GHz CPU), 192 Gb RAM and 2 × NVIDIA Tesla V100 GPUs – training took four days to converge at the final (512 × 512) resolution.

3. Results

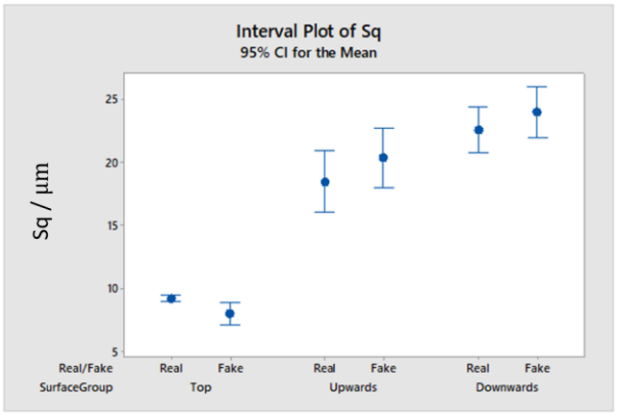
After the training procedure is concluded, the model was used to produce 1000 synthetic measurements from random points taken from the latent manifold. Figure 5 shows a susbet of the training data compared with a subset of the generator data. These are in unchanged form, so the height values are still encoded in the grayscale pixel values. The two sets are unique but, given an image of unknown origin, it would not be possible to distinguish whether it came from the generator or the training set on visual inspection alone. It can be seen that both sets include various surface types which correspond to the surface types that were shown in Figure 4. Top surfaces are easily distinguishable by the relative lack of particles and the distinct weld track lines. As the angle increases, the quantity of spatter particles increases. Figure 5 shows that the model can reproduce each of the surface types described in Figure 4.

Figure 6 shows un-encoded example measurements for a top surface, an upwards surface and a downwards surface. Each of these surfaces is compared to an equivalent generated surface taken from the model output. The accurate representation of surface defects can be seen clearly in Figures 6(d) and 6(f). Further defects and distinct weld tracks are visible in Figure 6(b). From the colour bars and through visual inspection of the surfaces in Figure 6, it can be seen that the scale of the features simulated are in line with the real data.

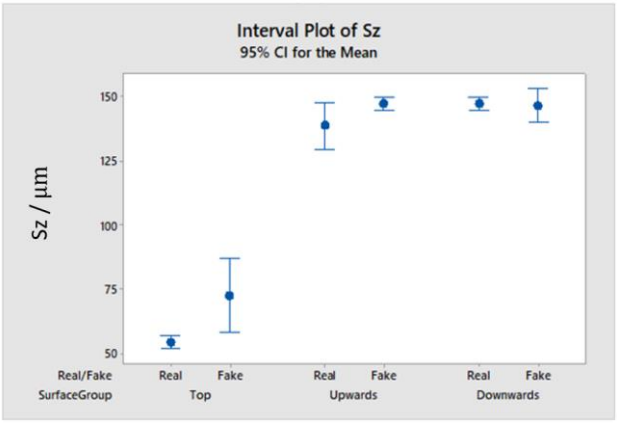
3.1 Statistical analysis of generated surfaces

ISO 25178-2 [12,13] defines a set of areal surface texture parameters for the analysis of surface topography mesurements. We can compare the distribution of these surface texture parameters across the entire training set and generated dataset. If the distribution of parameters is similar, this is an indication that the model produces generated measurements that are representative of real measurements of this kind. The parameters we compare are Sq and Sz, which are the root mean square and maximum heights of the scale limited surfaces

respectively. Figure 7 shows a comparison between the real and generated measurements of the distribution of the areal parameters for top, upward and downward surfaces.



(a)



(b)

Figure 7. Comparison of stastical ISO surface texture parameters for different surface types, showing the mean and 95% confidence interval for the entire training dataset (real) and the entire set of generated data (fake). (a) Sq, (b) Sz.

As can be seen in Figure 7, the distributions of real and generated measurements for each surface parameter overlap for upwards and downwards surfaces. The distributions match less consistently for top surfaces, which is likely because top surface measurements make up only 6% of the training data, whereas upwards and downwards surfaces each represent 47% of the training data, making it more difficult for the model to learn accurate representations of top surfaces. Figure 8 shows an example of an erronious top surface generated image that appears to show properties of both top and upward surfaces.

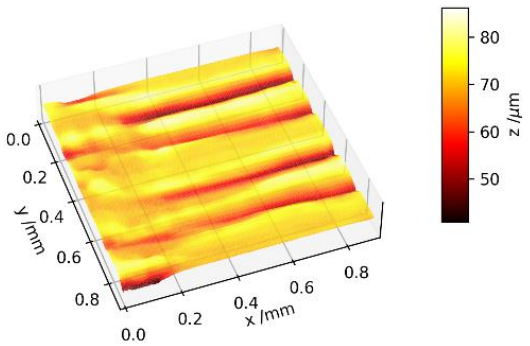


Figure 8. Generated surface which exhibits both top and upward surface features. This type of surface is not present in the training data.

Better representation of top surfaces could be achieved by increasing the proportion of top surface measurements present in the training data.

4. Future work

We have shown that the PG-GAN model can robustly produce synthetic surface measurements of different types. As the generated surfaces vary smoothly across the latent space there will be regions of this space that correspond to each surface type. Through analysis of the latent space, we should be able to learn which regions of this space generate each surface type and how "walking" in different directions across this space changes the various surface texture parameters. This would allow us to generate new data with predictable properties.

5. Conclusions

We have presented a method of generating large datsets of new and unique surface measurements that are representative of the kind of data one would expect to obtain through manual measurement. This model can produce surface texture parameters of different types while maintaining representative surface features. We have further shown reasonable statistical overlap in the distribution of the areal surface texture parameters between real and generated images for each surface type. While we use AM surfaces as a case study here, this model could easily be applied to surfaces of any type.

Acknowledgements

This work was supported by the EPSRC (grants EP/M008983/1 and EP/L016567/1) and Taraz Metrology Ltd.

References

[1] Liu M, Fai Cheung C, Senin N, Wang S, Su R, Leach R K 2020 On-machine surface defect detection using light scattering and deep learning *J. Opt. Soc. Am.* A **37** B53

[2] Gui J, Sun Z, Wen Y, Tao D, Ye J 2020 A review on generative adversarial networks: Algorithms, theory, and applications *arXiv:* 2001.06937

[3] Todhunter L, Leach R K, Lawes S, Blateyron F, Harris P M 2018 Development of mathematical reference standards for the validation of surface texture parameter calculation software *J. Phys. Conf. Ser.* **1065** 082004

[4] Lou S, Sun W, Brown S B, Pagani L, Zeng W, Jiang X, Scott P J 2018 Simulation for XCT and CMM measurements of additively manufactured surfaces *Proc. ASPE, Berkeley, USA* 189–194

[5] Li H, Li X, Chen Z, Liu X, Wang L, Rong Y 2018 The simulation of surface topography generation in multi-pass sanding processes through virtual belt and kinetics model *Int. J. Adv. Manuf. Technol.* **97** 2125–2140

[6] Li C, Xu K, Zhu J, Zhang B 2017 Generative adversarial nets *NIPS, Long Beach, USA* 4089–4099

[7] Ramachandran P, Zoph B, Le Q V 2017. Searching for activation functions *arXiv:1710.05941.*

[8] Arjovsky M, Chintala S, Bottou L 2017 Wasserstein generative adversarial networks *MLR, Sydney, Australia* **70** 214-223

[9] Karras T, Aila T, Laine S, Lehtinen J 2017 Progressive growing of gans for improved quality, stability, and variation *arxiv:* 1710.10196

[10] Newton L, Senin N, Chatzivagiannis E, Smith B, Leach R K 2020 Feature-based characterisation of Ti6Al4V electron beam powder bed fusion surfaces fabricated at different surface orientations *Addit. Manuf.* **35** 101273

[11] https://www.nottingham.ac.uk/it-services/research/uon-compute-service/

[12] ISO 25178-2 2012 Geometrical product specifications (GPS) — Surface texture: Areal — Part 2: Terms, definitions and surface texture parameters (Geneva: International Organization for Standardization)

[13] Leach R K 2014 *Characterisation of Areal Surface Texture* (Spinger)