Contents lists available at ScienceDirect

Precision Engineering

journal homepage: www.elsevier.com/locate/precision

Generation and categorisation of surface texture data using a modified progressively growing adversarial network

Joe Eastwood [*], Lewis Newton, Richard Leach, Samanta Piano

Manufacturing Metrology Team, Faculty of Engineering, University of Nottingham, UK

ARTICLE INFO

Keywords:
Optical metrology
Surface texture
Generative adversarial networks
Surface texture simulation
Machine learning
Focus variation microscope
Additive manufacturing

ABSTRACT

As machine learning becomes more popular in the precision engineering sector, the need for large datasets of measurement data increases. Due to the often manual, user dependent and labour-intensive measurement processes, collecting a large amount of data is often infeasible. It would, therefore, be desirable to collect a small amount of data on which to train a model to generate synthetic data that is representative of the real measurement data. To this end, we present an approach to numerical surface texture generation based on a progressively growing generative adversarial network. We show that by encoding height data into grayscale values within an image, the network can create realistic synthetic surface data both qualitatively and quantitatively. The proposed approach is general to any encoded surface; we demonstrate the model trained on two example datasets consisting of surfaces from different manufacturing processes and measured with different techniques. We finally present an extension to the generator model which automatically categorises the produced surfaces, allowing a surface of a desired category to be generated. Finally, we calculate the distributions of areal surface texture parameters for each type of surface and show that there is good agreement between the synthetic and real data.

1. Introduction

The ability to generate synthetic surface texture data which convincingly represents the result of a real measurement has many applications [1–3]. For example, often large quantities of data are required to train statistical models that would be difficult if not infeasible to collect manually [1]. Furthermore, representative synthetic textures are useful for other applications such as for use within virtual instruments [2], or for accurate image rendering [3]. Previous approaches for simulating surface data have been computationally intensive at runtime, limited to the representation of a single manufacturing process, or requiring an analytic representation of the surface [3–10]. Work by Eastwood et al. [3] used a synthetic surface texture of an additively manufactured (AM) part to produce photorealistic renders. These renders were then used to train a convolutional neural network (CNN) for object pose estimation. Here, the synthetic texture was simulated by analysing the real surface data of a part made with the same AM process, extracting the dominant spatial frequencies and amplitudes, and layering various pseudo-random noise functions at these frequencies and amplitudes. While this approach produces a good estimation of the surface parameters, it does not capture properties related to the surface features, such as feature shape and surface anisotropy. Software developed by Todhunter et al. [4] defined the surface to be simulated as a sum of cosine waves; the surface complexity can be increased further with the addition of pseudo-random noise in the form of multi-scale Fourier space Gaussian blur. This Fourier approach has some advantages as the generated surface parameters can be known explicitly, but the ‘realism’ of the generated surface is user-dependant. Another study used an analytic representation to generate surface form combined with smaller scale noise to simulate texture, resulting in a full synthetic topography [5]. The synthetic surface data was then used in the creation of synthetic interferometry data by phase-wrapping the simulated surface.

An alternate approach to realise synthetic surface data is to produce a full physics-based simulation of the manufacturing process of interest using numerical methods [6–8]. For example, Zhou et al. [9] focused specifically on the melt-pool of an arc-welding additive manufacturing (AM) process. Using a combination of a volume of fluid model and continuum surface force model to simulate both heat and mass transfer in the powder bed, they were able to predict the final surface profiles, which compare favourably with experimental data. This physics-based

[*] Corresponding author.
E-mail address: joe.eastwood@nottingham.ac.uk (J. Eastwood).

https://doi.org/10.1016/j.precisioneng.2021.10.020
Received 17 May 2021; Received in revised form 10 September 2021; Accepted 29 October 2021
Available online 2 November 2021

approach is wholly reliant on the accuracy of the physical simulation and is computationally expensive. Physical simulation models have the further disadvantage of being specific to a single manufacturing process; if surfaces are required to be simulated across a range of processes and materials, a large amount of development time would be required to develop new physical models.

Our proposed method overcomes many of the shortcomings of these previous approaches. Firstly, our model can simulate any process and measurement method that can be represented as a depth map and this same model can be retrained on a different dataset. Furthermore, the model can be trained to represent a range of surface types simultaneously (i.e. without the need for retraining) so long as the desired variation is represented within the training data. Moreover, we show that the produced surfaces are representative of real data without the need for manual analysis of the desired surface features.

In this paper we present a new method to produce surface texture data based on an approach initially developed for generating high resolution synthetic images: a progressively growing adversarial network (PGGAN) [10]. By encoding training measurements as high-resolution images with height data represented by the pixel values, we can create a dataset to train the PGGAN. Once the model is trained, it will generate images with the same encoding, which can then be decoded back into height data. By front loading the computational expense to training time, once deployed, the model can quickly generate large quantities of new surface data. Further, we show that a single model can simulate a variety of surface types simultaneously and then extend the PGGAN model to automatically categorise the generated surfaces into predefined surface types. The performance of the proposed method is validated on two very different datasets: a collection of industrial coatings measured using fringe projection measurement and an AM part measured using focus variation microscopy. We then extend the surface generation model to perform a categorisation of the generated surface types creating a model that can produce surfaces with predictable properties. Finally, we perform a quantitative comparison of the categorised generated surfaces with their real counterparts and show the model provides a sound representation of the surface types.

2. Surface generation model

A generative adversarial network (GAN) is a system of two subnetworks trained in a zero-sum-game (first proposed in 2014 by Goodfellow et al. [11]). Given some set of input data, the task is to generate some new data that cannot be distinguished from the original dataset, while capturing the variation present within the original data. To achieve this task, a sub-network, called the generator ($G(\mathbf{z}) \rightarrow \mathbf{i}$) takes some input vector as a seed value from which it produces data of the same type as the input (for example, an image). Initially, the generator output is pseudo-random over the input. The second sub-network, called the discriminator ($D(\mathbf{i}) \rightarrow p$), uses data $\mathbf{i}$ sampled from $G(\mathbf{z})$, or taken from the initial dataset, and produces a prediction p as to whether $\mathbf{i}$ is 'real' (from the dataset) or 'fake' (from the generator). The training is a zero-sum-game because the generator loss function is low when the discriminator loss function is high, i.e., when the generator successfully tricks the discriminator into believing some generated data is real. In the original publication [11], both G and D were differentiable functions represented by multi-layer perceptrons; however, it is now more common to use convolutional layers, especially when dealing with images, as is the case in this work. A generic convolutional GAN architecture is shown in Fig. 1.

A PGGAN is an extension of the traditional GAN architecture that was originally proposed by NVIDIA [10]. A PGGAN improves variability and stability when operating on high resolution images by beginning with a highly down-sampled version of the training data, in our case (4 × 4) pixels. After a predefined number of training periods (epochs), an additional transpose convolution layer is appended to the generator model and a conventional convolution layer is prepended to the discriminator, doubling the resolution of the generated image. The resolution doubling is repeated until the final resolution is achieved (in this case (512 × 512) pixels). The additional layers are faded-in to the model smoothly over a period of epochs to avoid any jerk to the network and encourage stability; this is shown in Fig. 2.

This smooth fading is achieved by adding a 2 × up-sampling layer to the generator and a 2 × down-sampling layer to the discriminator. The output of the new convolutional layers is combined in a weighted sum with the up-sampling/down-sampling output, where the relative weighting of each contribution is controlled by a parameter α. Over a predefined number of epochs (where this number is a hyper-parameter of the PGGAN model) the weighting parameter linearly increases until the up-sampling/down-sampling layers no longer contribute to the model and can be removed.

The input vectors to the generator are sampled from a 'latent space'; in our case this space is the unit hypersphere S^{99} which is defined by,

$$S^{99} = \{\mathbf{x} \in \mathbb{R}^{100} \mid \|\mathbf{x}\| = 1\}. \tag{1}$$

In the case of the PGGAN, the discriminator, rather than classifying the input as either real or fake, assigns a continuous 'realness' value to the input. Using a continuous realness value rather than discrete classification supplies a smoother gradient and leads to more stable training

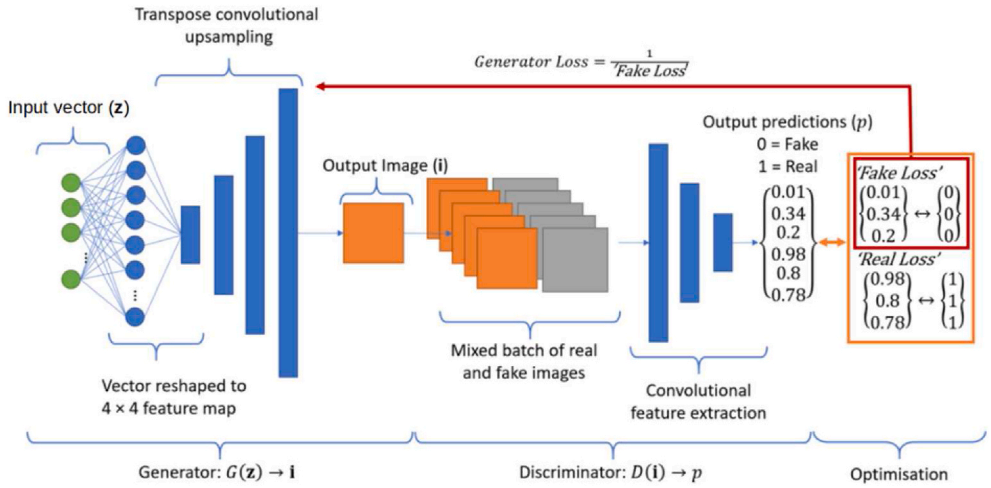

Fig. 1. A generic GAN showing the generator model, the discriminator model and the zero-sum optimisation through which they are trained. A variety of loss functions are available for calculating the real and fake losses but it is common to use the binary cross-entropy.

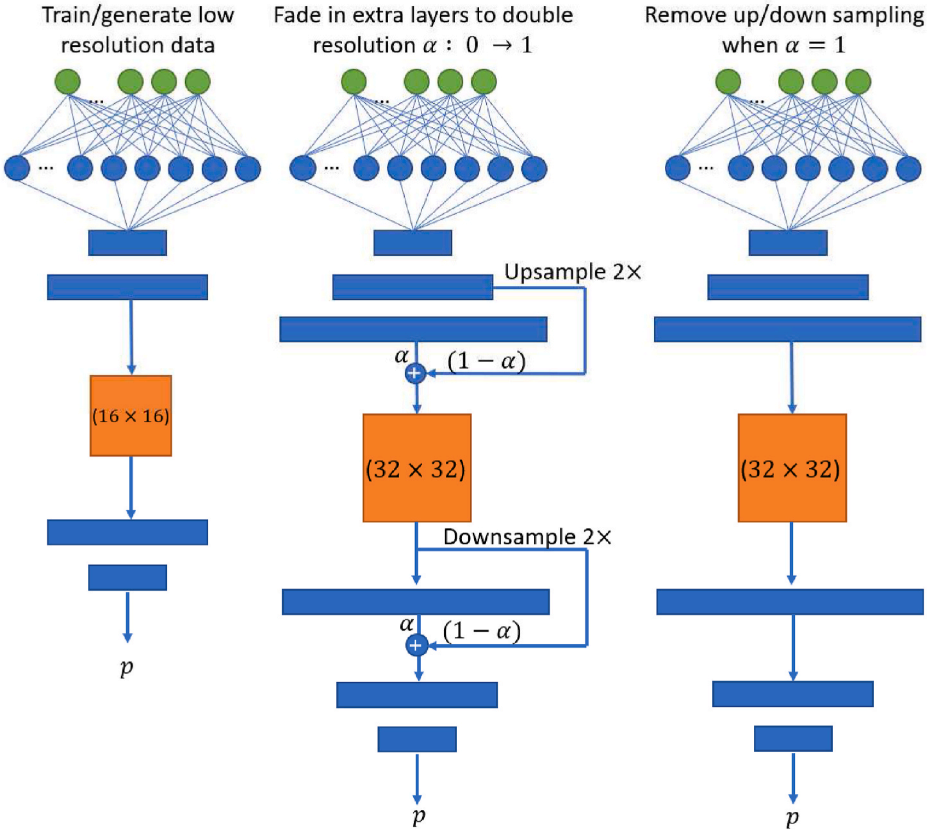
Train/generate low resolution data
Fade in extra layers to double resolution α : 0 → 1
Remove up/down sampling when $\alpha = 1$
...
Upsample 2×
α
$(1-\alpha)$
(16 × 16)
(32 × 32)
(32 × 32)
Downsample 2×
α
$(1-\alpha)$
p
p
p

Fig. 2. The process of doubling the resolution smoothly in a PGGAN using the α parameter.

[12]. In turn, these realness prediction values are fed into a loss function based on the Wasserstein distance, a measure of the minimum amount of work required to turn one distribution into another [13]; in the case of a GAN, this is the distribution of the critic predictions compared to the real distribution of real/fake images. A Wasserstein based loss prevents vanishing and exploding gradients when compared to cross-entropy approaches (which is the popular alternative).

Once the PGGAN is trained and the discriminator is discarded, the generator can be used to generate new images that have been shown to be indistinguishable from the real dataset discriminator. For our application, we extend the generator by piping the output of the generator into the input of a CNN to classify the type of surface produced. This process allows us to make meaningful *comparisons* of statistical surface texture parameters to ensure that the synthetic surfaces are representative of the full space of real measured surfaces. Full details of the CNN extension are given in section 6.

3. Datasets

Two datasets were developed to train, validate and test the model: industrial coatings and AM surfaces. To show that the approach is applicable to a range of measurement techniques and surface types, the datasets described below use different measurement techniques on different surface types. In both cases, the same procedure was used to convert the measured data into the final set of (1024 × 1024) pixel images. The measured data were first converted into depth maps before a polynomial form removal was applied. The height data were then encoded as a set of grayscale images. A process of dataset augmentation was used to expand the datasets. This process involved rotating, mirroring and cropping the images to a size of (512 × 512) pixels.

3.1. *Industrial coatings*

A set of sample surfaces created from a variety of industrial coatings were produced which were then measured using a fringe projection surface texture measurement system [14]. The procedure for data treatment described above was applied; in this case each encoded image represents a (20 × 20) mm area and depth values are encoded as a linear mapping to grayscale values in the range (0–1) from depth values in the range (−50 to 50) μm. A sample of the final treated data is shown in Fig. 3. The industrial coating surfaces were considered a suitable case study because there are various combinations of process parameters that can create a large range of resultant surfaces, however, there is a fundamental limitation on how many surfaces could be economically produced – making the ability to simulate the potential “design-space” of all possible surfaces a valuable endeavour.

3.2. *AM surfaces*

Another dataset was constructed from focus variation microscope measurements of a metal AM part [15]. The ‘bracelet’ artefact consists of a series of thirty-six plane faces at 10° increments with minimal supports produced by electron beam powder bed fusion (EBPBF) [16]. This artefact was chosen as it will give a range of surface types dependent on the relative orientation of the face to the powder bed. The CAD of this part is shown in Fig. 4(a).

Fig. 4(c) shows the part manufactured from Ti6Al4V using an Arcam A2X EBPBF process. These data were collected using a focus variation microscope with 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. The data processing steps outlined previously were applied, this time mapping depth values of (−70 to 70) μm to grayscale (0–1) and the image size representing (1 × 1) mm. Fig. 5

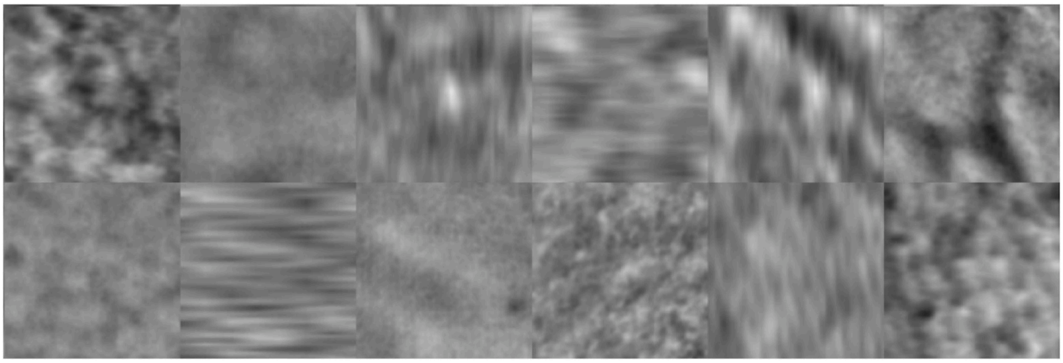

Fig. 3. Twelve example encoded images taken from the industrial coatings dataset, showing the range of different surface types present in the training data.

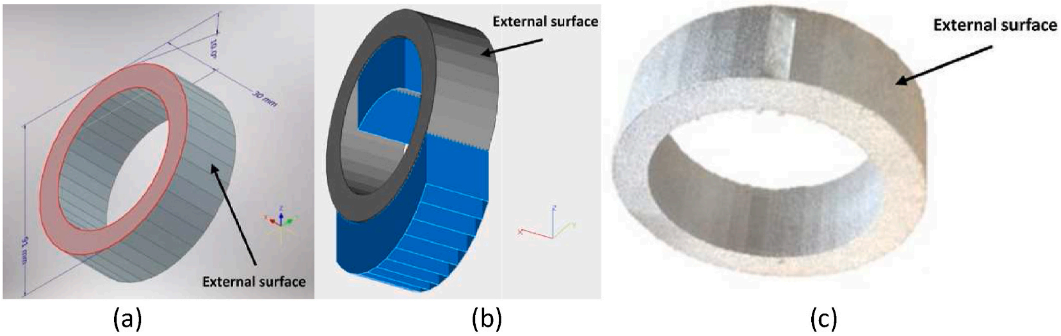

Fig. 4. (a) Bracelet artefact CAD model, (b) bracelet artefact support structures, (c) photograph of manufactured artefact.

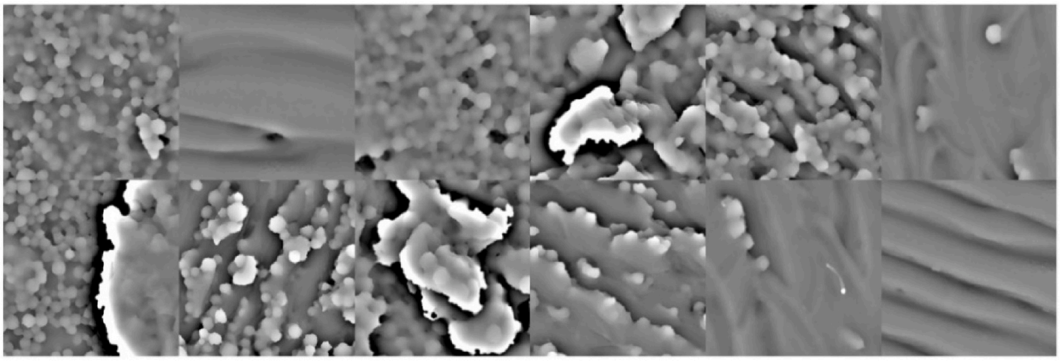

Fig. 5. Twelve example encoded images taken from the AM dataset showing the large variation present between surfaces included the dataset.

shows a sample of the measured data with the post processing steps applied.

It is clear that there is a large variation in the types of surfaces measured from the part. This variation is dependent on the relative angle of the measured face to the powder bed [15]. For example, the large-scale features that can be seen in Fig. 5 are the remnants of the support structures shown in Fig. 4(b), which were required for the printing process and then removed post-process. These support structures only occur on the down-skin surfaces. Additionally, the smooth, straight weld tracks only occur on the top face, which is parallel to the powder bed. As the surface angle increases relative to the build plane, the presence of particles agglomerated to the surfaces increases, eventually occluding the weld tracks entirely. Fig. 6 shows these surface types, their location on the artefact, and an example from the final dataset of each type. Section 6 discusses these surface types in more detail and how the generator can be extended to produce surfaces of a known type.

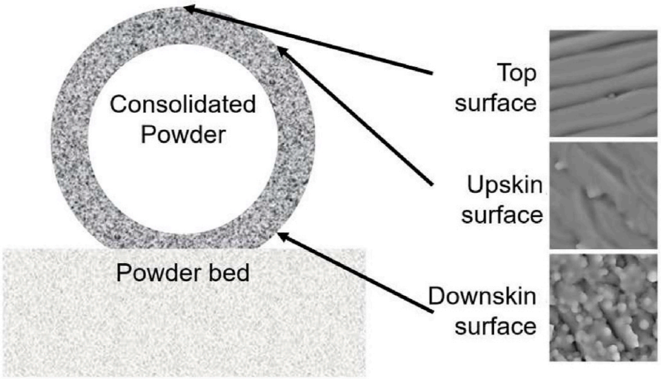

Fig. 6. Surface types and their locations on the bracelet artefact with examples taken from the AM surface dataset.

4. Surface generator results

For both datasets, the PGGAN took five days to complete training on a high-performance computing (HPC) graphics processing unit (GPU) node on the Augusta HPC at the University of Nottingham [17]. The specifications of the HPC node used were: two Intel Xeon Gold 6138 20C 2.0 GHz CPU, 192 Gb RAM, and two NVIDIA Tesla V100 GPUs. The length of the training time is in agreement with the original PGGAN paper [10] for producing images of a similar resolution and once trained the model can produce new surface texture data in less than a second. Once training had concluded, the trained generator model was deployed to create 1000 images of both the coated surfaces and the AM surfaces. Fig. 7 shows a comparison of the real and synthetic images for the industrially coated dataset.

The examples shown in Fig. 7(a) were selected to show the range of possible surface data contained within the training dataset. It can be seen in Fig. 7(b) that this variation is captured by the generator with considerably different features visibly present across the output data. Fig. 8 shows a similar comparison for the AM dataset.

The model outputs still need to be decoded from grayscale images into true height data. To do this, the reverse of the encoding process described previously is applied to the 1000 generated images. Fig. 9 shows example decoded surfaces compared with real surfaces of the same type from the training data.

As can be seen in the scale of Fig. 9, the heights generated and scales of features generated match closely with those in the training data. Fig. 9 (d) shows the model has learned to represent defects in the weld tracks which indicates this method could be useful for training defect detection models - a common issue within the field (see review by Meng et al. [18]). Simulated surfaces that possess features found in the real surface

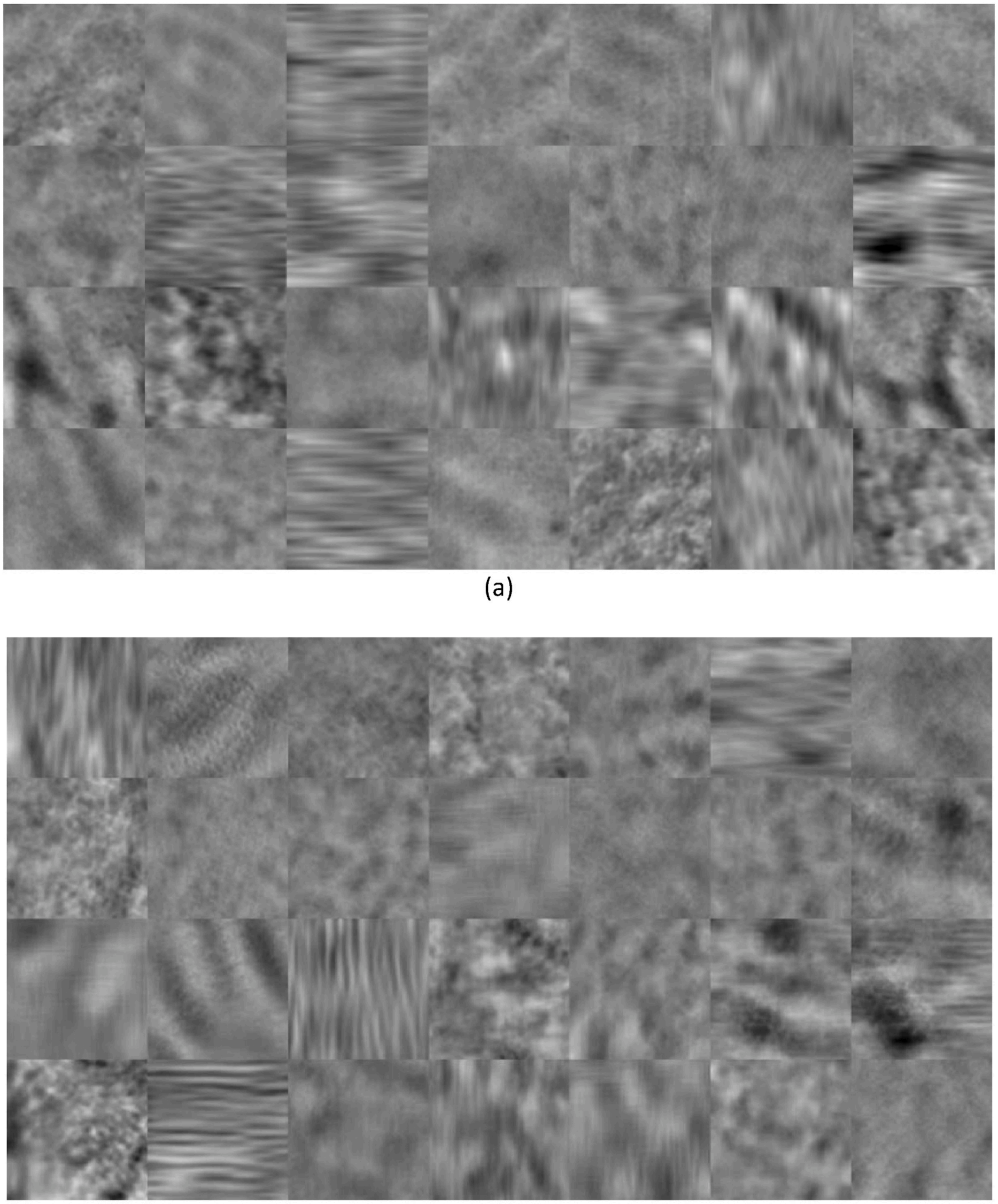

Fig. 7. A random sample of twenty-eight surface images taken from (a) the real encoded depth data from the industrial coating dataset, (b) the synthetic encoded depth data created by the trained PGGAN generator.

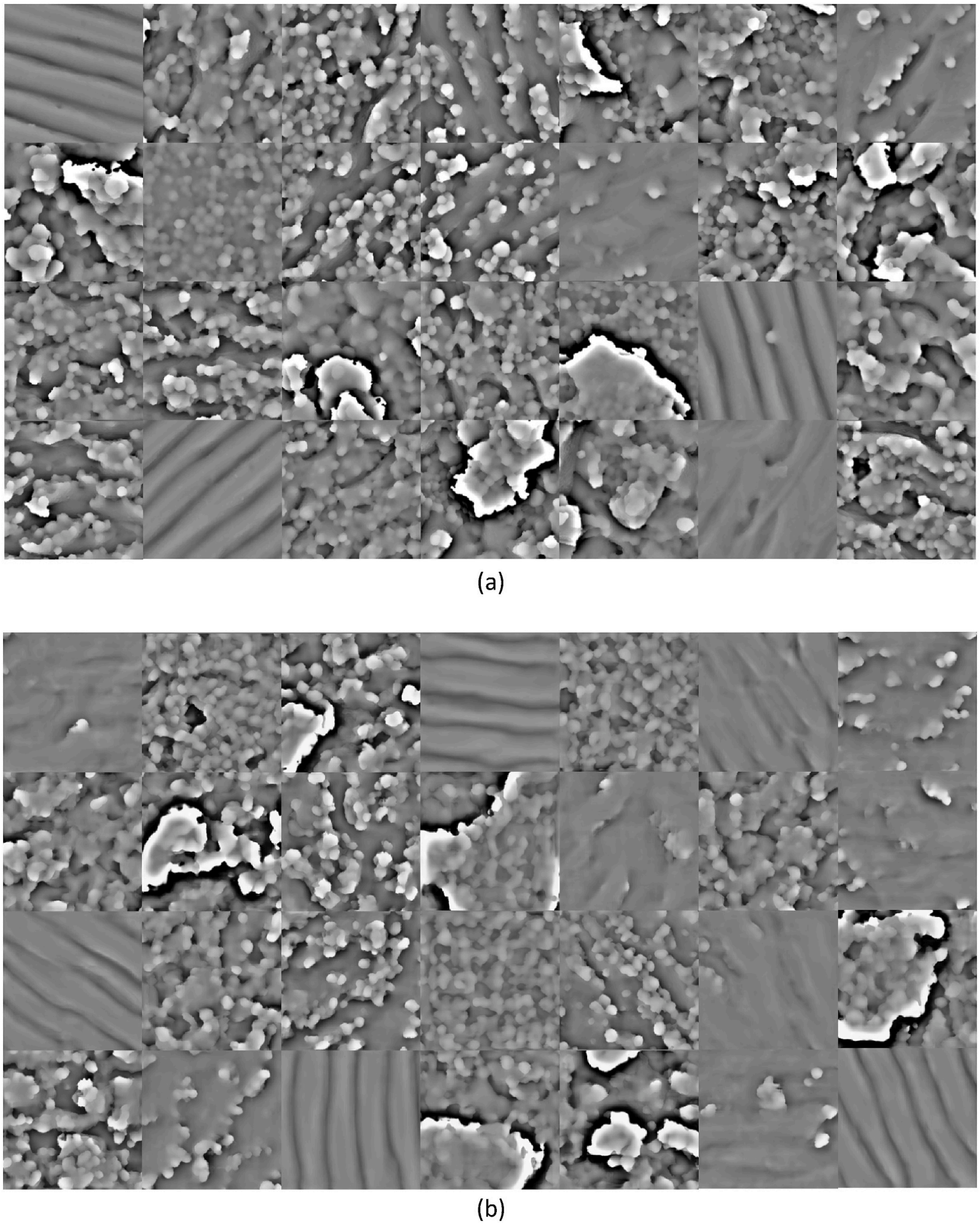

Fig. 8. A random sample of twenty-eight surface images taken from (a) the real encoded depth data from the AM dataset, (b) the synthetic encoded depth data created by the trained PGGAN generator.

case could also allow for more training data for the development of improved feature-based characterisation approaches.

5. Generating categorised surfaces

The trained model generates surfaces by randomly sampling a point from S^{99} and passing the input coordinate through the generator, producing a randomly sampled surface from the possible output-space. As discussed in section 4, in the case of the AM dataset specifically it is clear that there are distinct types of surfaces encapsulated by the dataset, which were shown in Fig. 6. Top surfaces are characterised by distinct weld tracks and the absence of agglomerated particles, top surfaces are produced when the face is parallel with the powder bed. In up-skin surfaces, the weld tracks can still be seen but, as the angle relative to the powder bed approaches 90°, particle agglomeration begins to dominate. Finally, down-skin surfaces are fully dominated by agglomerated particles due to the increased interaction with the powder bed and the remnants of support structures (from those shown in Fig. 4(b)) can sometimes be seen.

Given these categories, the model would be more useful if it could be used to generate surfaces of a known type rather than a random surface. To this end, we extend the generator model by piping the output directly into the input of a secondary CNN that predicts the generated surface type. The categorisation CNN architecture is shown in Fig. 10.

Moreover, we can use the same AM surface training set used for the PGGAN to train the categorisation model. During the measurement, the angle of the face being measured was recorded and stored in the metadata associated with the measurement data. It is a straightforward

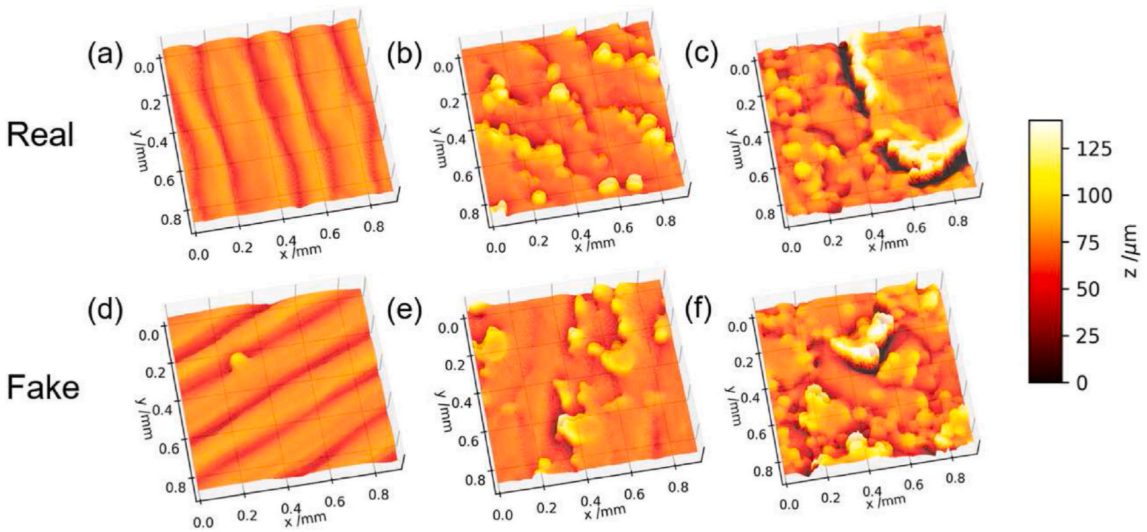

Fig. 9. Surface topography height maps of real measurement data from the AM surface dataset compared to decoded fake data from the generator.

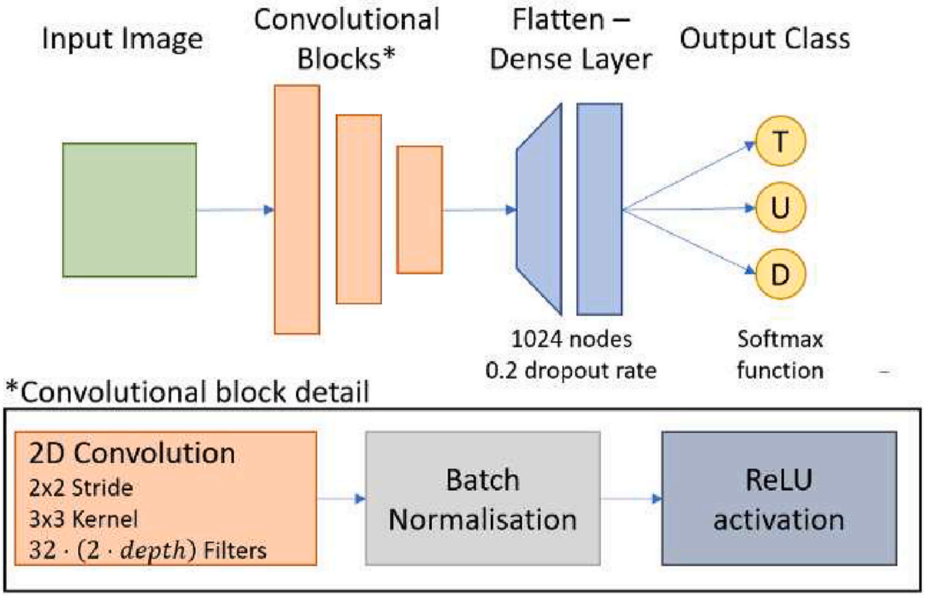

Fig. 10. Categorisation CNN, which takes an input image, extracts features through a sliding window kernel convolution, flattens the feature maps, feeds through a fully connected layer and produces a predicted class label (T: top,U: up-skin, D: down-skin).

process, therefore, to propagate this metadata through the data augmentation process to supply ground-truth labels. Measurement data originating from the 0° face were labeled as top, data from faces in the 10°–90° interval were labeled as up-skin, and 100°–180° as down-skin. There is some ambiguity as to whether the 90° face should be categorised as up-skin or down-skin and, because the physical characteristics transition smoothly between these two categories, there is likely to be some misclassification of surfaces near the boundary. For other datasets it may be optimal to set the boundaries at different angles, due to the effect of gravity during processing for example, however it was found that the classification in this case was most accurate when using the boundaries detailed above.

A softmax function (a normalised exponential function which can be thought of as a generalisation of the logistic function to *n*-dimensions [19]) was used as the activation function in the output layer of the model. The Adam optimiser [20] with a learning rate of 0.0001 with a categorical cross-entropy loss function were used in the training of the CNN. Due to the relative simplicity of the categorisation model when compared to the PGGAN model, a HPC compute node was used for training, which was completed within 12 h. Fig. 11 shows a plot of the training history. The loss values shown are the values of the cross-entropy over that image batch; the accuracy is calculated by simply taking the argmax (the index of the output tensor containing the maximum value) of the values of the output nodes.

As can be seen in Fig. 11, the validation loss reached a minimum at around 1000 training steps (batches of sixty-four images) after which overfitting began to occur. To prevent overfitting, two mitigation strategies were employed. The first strategy was to use an early stopping criterion which monitored validation accuracy and ceased the training procedure if no improvement was observed within ten epochs. The second strategy was to use a model checkpointing system which, once training is finished, restores the model weights to the point at which validation accuracy was a maximum. In this case, the maximum validation accuracy achieved was 96%.

When deployed, rather than simply taking the argmax of the output nodes to perform the categorisation, a 'certainty threshold' was set at

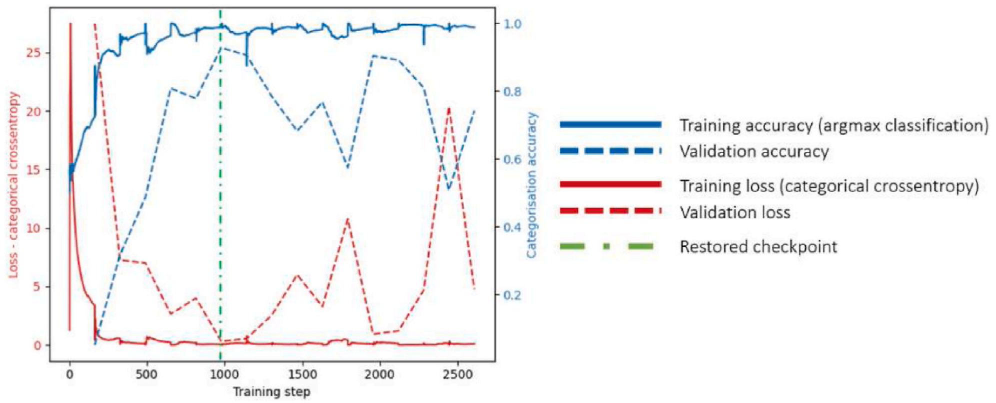

Fig. 11. Categorisation CNN training history. To prevent overfitting, ten percent of the dataset was used for cross validation and model weights restored to the maximum validation accuracy.

80%. That is to say, the input was only assigned to the predicted class if the value of the corresponding output node was larger than 80%; if this condition was not met, the image was instead categorised as 'uncertain'.

In order to train on the labeled real data, the categorisation operation can be performed on the 1000 generated surfaces (a sub-sample of which is shown in Fig. 7(b)). Of these 1000 images, 4.6% were classified as uncertain. This level of uncertainty is an indication that the generated data does in fact accurately capture the input space as the rate of uncertainty correlates with the misclassification rate of the model during validation. When inspecting which surfaces are misclassified, the majority bear similarities to the surfaces around the 90° angle relative to the powder bed, as was expected. An example of this is shown in Fig. 12.

Forty-three of the forty-six 'uncertain' images fell into the category shown in Fig. 12; the remaining three images had a different failure mode. These generated surfaces are classed as uncertain because they are not representative of the surfaces contained in the training data. Specifically in this case, they are all surfaces that have the distinctive weld track features of the top surfaces but also the large amount of particle agglomeration of the other surface types. Fig. 13 shows an example of this surface.

As the unrepresentative images, such as the example shown in Fig. 13, occur at such a low rate (0.3% in this test) and we have now trained a model to sort them from the generated images which are representative of the real data, we can simply discard these surfaces and consider only the successfully categorised surfaces in our analysis.

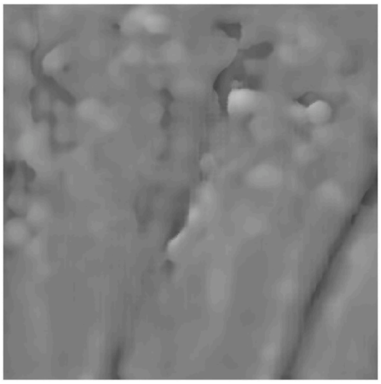

Fig. 13. An unrepresentative image showing encoded depth data produced by the generator showing clear weld tracks and noise from agglomerated particles.

6. Quantitative comparisons

A benefit of our extended model is that we can compare surface statistics between the surface categories independently rather than averaged statistics for the complete dataset. As these surfaces have such different features, this method will provide a much more robust analysis than without this extension. First we consider parameters based on the surface height distribution relative to the mean plane: *Sq* is defined as the root-mean-square height deviation and *Sz* is simply the maximum

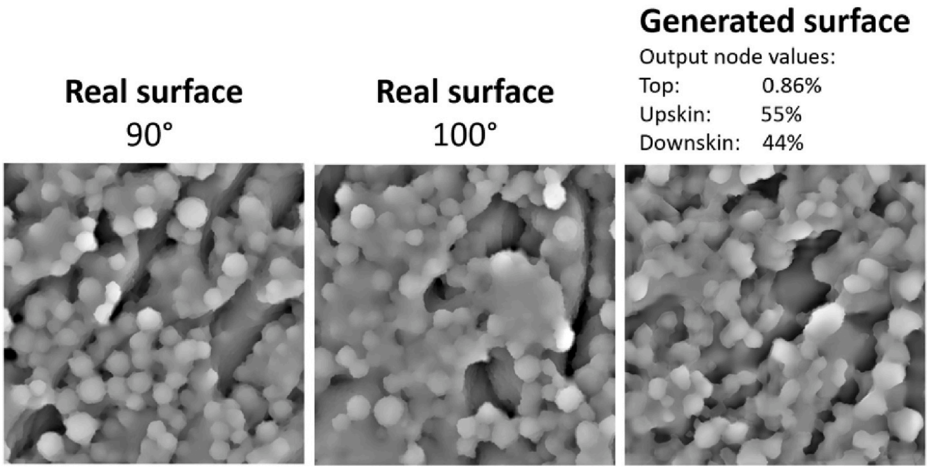

Fig. 12. Comparison of encoded depth data for an example of a common misclassified surface with real surfaces around the borderline of the up-skin/down-skin categories.

height [21,22]. The distributions of these two parameters are shown in Fig. 14 for real and simulated surfaces across each surface category.

As can be seen, the distributions produced by the generator show good agreement with the training data. While the comparison of height parameters is useful to begin to show good representation, it is not the full picture, for example, two very different surfaces could have similar *Sz* values. Considering spatial parameters in addition to amplitude parameters can provide a more complete comparison. Spatial parameters describe properties related to the distributions of the shape and size of the features that make up the surface texture. In this case, we compare three parameters: *Sal*, amplitude of the dominant spatial wavelength and dominant spatial wavelength. *Sal* is the fastest decay autocorrelation length, which is a measure of the distance from given point on the surface to a point which has minimal correlation with the starting point [21,22]. The distributions of these three parameters are given in Fig. 15.

In both Figs. 14 and 15 the synthetic and real surface parameter distributions overlap in most cases or are different by small absolute amounts. For example, Fig. 15(a) shows the mean *Sal* of top surfaces differs by only two microns despite the distributions not overlapping. It makes sense that the top surfaces are less well represented than the other surface categories as they make up a relatively smaller proportion of the dataset (only 5% of the input data were top surfaces compared to 48.2% for the remaining categories), this is simply because there were fewer top surfaces to measure on the ring artefact. This is evidence that the synthetic surfaces are not only qualitatively similar to the real data but quantitatively similar, and that any differences are small. This similarity shows that the output space of the model is at least partially representative of the input surfaces. It is interesting to note that, particularly for the spatial parameters, the distributions of the generated surfaces are much tighter than the distributions present in the training data. This tighter spread is likely due to the generator learning to represent a subset of the input space - this is discussed in section 7.

7. Discussion

We selected the PGGAN model over other generative methods for two main reasons. Firstly, it has been developed specifically to encourage stability and variation in the generator outputs at high resolution. This means that the model is likely to learn to represent a larger portion of the input space than competing methods at the resolution of data within our example datasets. Secondly, recent variations of the PGGAN have been developed for more specific applications, such as style transfer (see StyleGAN [23] and Cycle-GAN [24]), whereas the original PGGAN implementation can be applied generically to any input image dataset.

As was noted at the end of section 6, the distributions of areal surface texture parameters show good agreement between the real and generated surface but do not match exactly. Firstly, the variance in the generated data is, in general, smaller than the variance among the training data. It is a known shortcoming of GANs that commonly only a subset of the possible variation is represented by the trained model [25]. This is intuitive, as it is simpler for the model to learn to represent a subset of the input space to a high enough quality to trick the discriminator than to learn to represent the entire space. As stated previously, many of the features of the PGGAN are specifically designed to increase variation in the output (this is discussed at length in the original PGGAN paper [10]) but, at least for our application, there is still some work to do in this area.

A feature of using a machine learning approach is that the model will learn to represent patterns that are present within the training data. This means that any measurement errors present in the training data will be replicated in the synthetic surfaces. Replicating measurement errors could be disadvantageous to some applications where it is desirable to produce simulations of true surface topographies, however, it is an advantage if the application calls for simulated measurement data from a real instrument. This method has been shown to be effective at generating both fringe projection and focus variation measurement data, however, it is general to any technique where the data can be represented by a height map. Fringe projection surface measurement operates at larger scales than most optical texture instruments and focus variation has relatively low resolution when compared to a technique such as coherence scanning interferometry, and these characteristics are replicated in the generated data. If the proposed approach was applied to another measurement technique, we would expect limitations inherent to that technique to be reproduced by the trained model.

Encoding the training data as a grayscale image does not affect the spatial resolution of the training data as each measured surface point is represented by a unique pixel value and the data is stored in a lossless format to ensure no compression artefacts are introduced. However, one shortcoming of the method is that the vertical resolution of the model is limited to 255 discrete pixel values. In the case of the AM dataset this introduces an uncertainty of ±0.25 μm when encoding the input data. This is close to the 0.1 μm spacing between stacked focus variation images so unlikely to have a large effect on the data quality. Replacing the input encoded image with a floating-point array would effectively eliminate these errors.

8. Conclusions

We have presented a novel approach to the generation of synthetic surface data by exploiting an approach initially designed for the synthesis of high-resolution images. We show that by encoding the surface height data into the grayscale channel of an image, we can train a PGGAN model to produce new data that represents a training set of images. By applying a process of dataset augmentation, we make the model robust to some transformations, such as rotation, and allow the

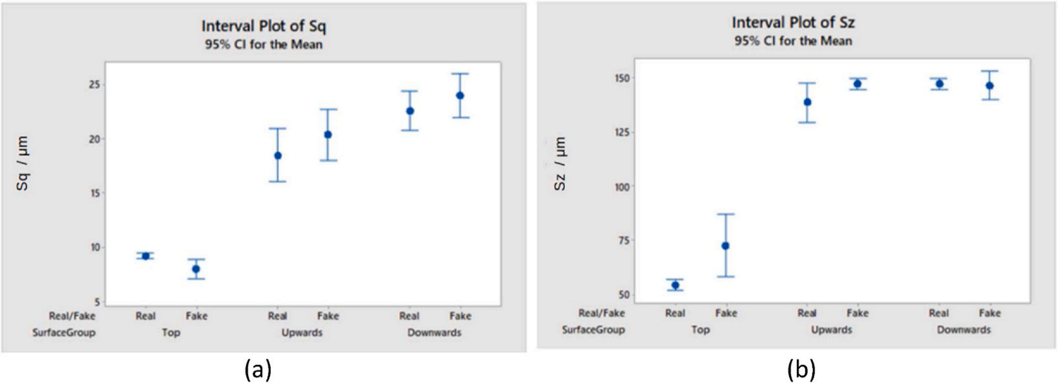

Fig. 14. Comparison of the mean and 95% confidence interval of the spreads of (a) *Sq* and (b) *Sz* r for each surface category, for real surfaces in the AM surface dataset and generated surfaces from the PGGAN generator.

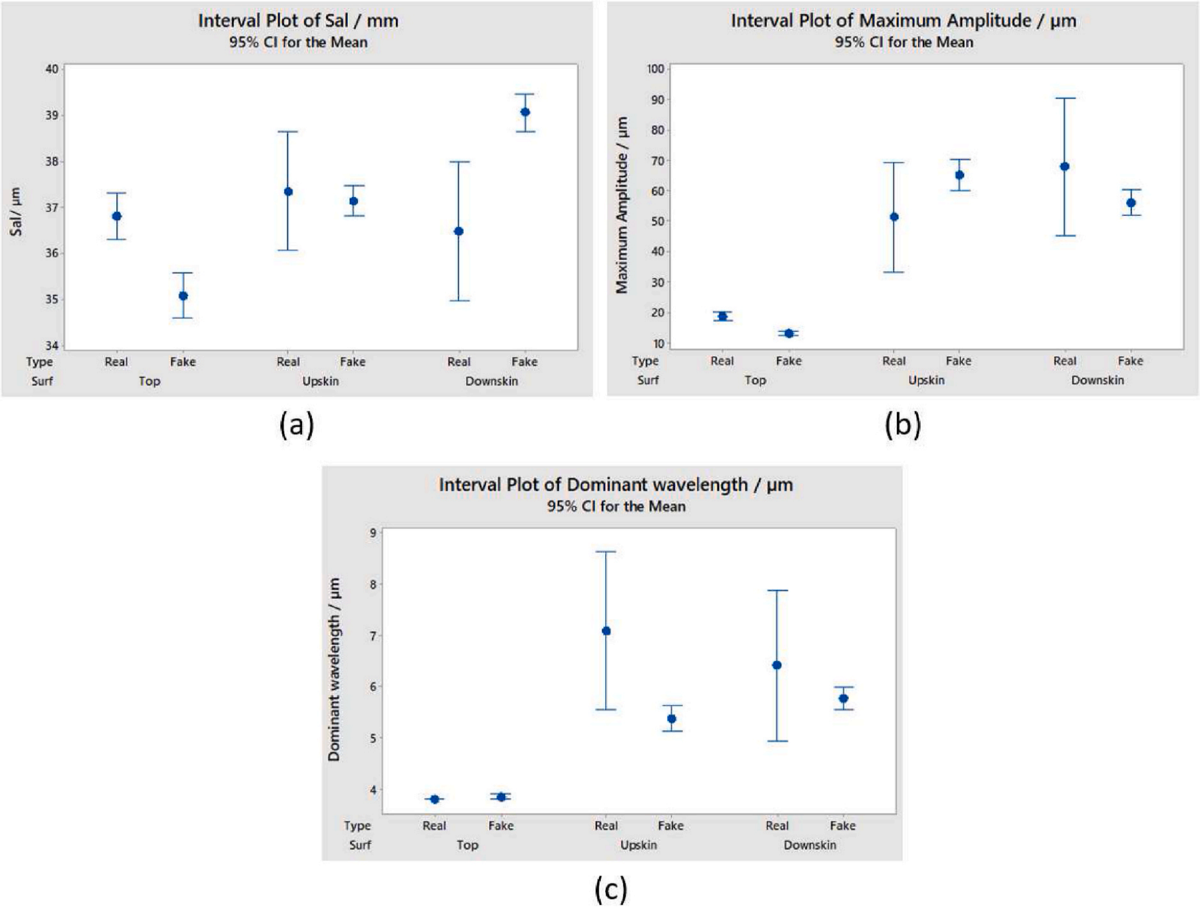

Fig. 15. Comparison of spatial parameters (a) *Sal*, (b) amplitude of the dominant spatial wavelength, (c) dominant spatial wavelength, for each surface type for surfaces from the AM surface dataset and surfaces produced by the PGGAN generator, showing the mean and 95% confidence interval in each case.

initial measured dataset to be relatively small (less than 100 measurements). We show that a CNN can be used to categorise the surfaces produced into categories that are known in the initial dataset. This categorisation allows the model to produce surfaces of a desired type, rather than a random sample from the space of all possible surfaces that the model can represent. Furthermore, this categorisation allows for specific comparisons between the distribution of areal surface texture parameters over the categories of surfaces, rather than the full datasets.

We present two case study datasets, one derived from fringe projection measurements of industrially coated surfaces and the other from focus variation measurements of metal AM surfaces. We show in both cases that the generated surfaces are visually similar to those in the original dataset. In the case of the AM surface, we show that our approach successfully classifies 96% of the unseen data. The 4% of the data classified as ‘uncertain’ was due to fuzzy boundaries between the up-skin and down-skin categories and a small number of generated images that poorly represented the training data (0.3%). Finally, a quantitative analysis of both amplitude and spatial areal surface texture parameters was conducted. The distributions of these parameters for the synthetic data shows relatively good agreement with the distributions of the real data. There is an indication, due to the tighter distributions in the synthetic data, that only a subset of the possible real surfaces have been represented by the generator model. This lack of variation is a known shortcoming of the GAN and although the PGGAN takes steps to increase variation in the generator, in the case of our AM surface data at least, this is an open issue.

As the surfaces used have been shown to be quantifiably representative of those within the training data, large quantities of synthetic surface data can be produced quickly to go on to be used in a variety of possible applications including, but not limited to, training statistical models, virtual instruments, and accurate surface simulation and rendering.

9. Future work

A simple further next step in analysis of this work would be to consider hybrid parameters such as *Sdr* (a measure of total developed area of all tessellations) and *Sdq* (mean quadratic slope) which could provide further insight into the synthetic surfaces.

Taking the model further will include the use of principle component analysis (PCA) on the early activation layers of the generator model to map the latent space. An implementation of PCA on similar models has been presented recently (called GANSpace [26]) and has been shown to allow the development of semantic control over the generator output. For the application to surface texture, this could allow the generation of surfaces with prescribed properties. An area of particular interest is to generate interpretable controls for creating synthetic surfaces representative of those which would be produced through a specific combination of process parameters. Additionally, further work refining the model architecture to be more performant specifically on datasets of the form presented here could yield generator models with greater stability and variation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the EPSRC (grants EP/M008983/1 and EP/L016567/1) and Taraz Metrology Ltd. We thank the HPC team at the University of Nottingham for use of the Augusta HPC on which this work was completed.

References

[1] Tao X, Zhang D, Ma W, Liu X, Xu D. Automatic metallic surface defect detection and recognition with convolutional neural networks. Appl Sci 2018;8:1575.
[2] Su R, Leach RK. Physics-based virtual coherence scanning interferometer for surface measurement. Light Adv. Manuf. 2021;2:1–6.
[3] Eastwood J, Sims-Waterhouse D, Piano S, Wier R, Leach RK. Pose estimation from a monocular image for automated photogrammetry *Proc*. Geneva, Switzerland): euspen Int. Conf.; 2020.
[4] Todhunter L, Senin N, Leach RK, Lawes S, Blateyron F, Harris P. A programmable software framework for the generation of simulated surface topography. Proc. euspen Int. Conf. (Venice, Italy) 2018.
[5] Pineda J, Altamar Mercado H, Romero L, Marrugo A. Toward the generation of reproducible synthetic surface data in optical metrology *Proc*. SPIE 11397; 2020. 113970C.
[6] Cheung CF, Lee WB. Modelling and simulation of surface topography in ultra-precision diamond turning. Proc. Inst. Mech. Eng. B J. Eng. Manuf. 2000;214: 463–80.
[7] Gao T, Zhang W, Qiu K, Wan M. Numerical simulation of machined surface topography and roughness in milling process. J Manuf Sci Eng 2006;128:96–103.
[8] Reizer R, Galda L, Dzierwa A, Pawlus P. Simulation of textured surface topography during a low wear process. Tribol Int 2011;44:1309–19.
[9] Zhou X, Zhang H, Wang G, Bai X. Three-dimensional numerical simulation of arc and metal transport in arc welding based additive manufacturing. Int J Heat Mass Tran 2016;103:521–37.
[10] Karras T, Aila T, Laine S, Lehtinen J 2018 Progressive growing of GANs for improved quality, stability, and variation Proc. ICLR (Vancouver, Canada).
[11] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. *arXiv* 1406.2661. 2014.
[12] Jolicoeur-Martineau A. The relativistic discriminator: a key element missing from standard GAN. 2018. *arXiv* 1807.00734.
[13] Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks *Proc*. Sydney, Australia: ICML; 2017. p. 214–23.
[14] Dickins A, Widjanarko T, Sims-Waterhouse D, Thompson A, Lawes S, Senin N, Leach RK. Multi-view fringe projection system for surface topography measurement during metal powder bed fusion. J. Opt. Soc. Am. A. 2020;37: B93–105.
[15] Newton L, Senin N, Chatzivagiannis E, Smith B, Leach RK. Feature-based characterisation of Ti6Al4V electron beam powder bed fusion surfaces fabricated at different surface orientations. Addit. Manuf. 2020;35:101273.
[16] Körner C. Additive manufacturing of metallic components by selective electron beam melting—a review. Int Mater Rev 2016;61:361–77.
[17] https://www.nottingham.ac.uk/it-services/research/uon-compute-service/.
[18] Meng L, McWilliams B, Jarosinski W, Park HY, Jung YG, Lee J, Zhang J. Machine learning in additive manufacturing: a review. J. Micros. 2020;72:2363–77.
[19] Goodfellow I, Bengio Y, Courville A. Softmax units for multinoulli output distributions. In: Deep learning. Cambridge: MiT Press; 2016. p. 180–4.
[20] Kingma D P, Ba J 2015 Adam: A method for stochastic optimization *Proc*. ICLR (San Diego, USA).
[21] ISO 25178 part 2 2012 Geometrical product specifications (GPS) — surface texture: areal — Part 2: terms, definitions and surface texture parameters (Geneva: International Organization for Standardization).
[22] Leach RK. Characterisation of areal surface texture. Springer; 2014.
[23] Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. Proc. CVPR (Long Beach, USA) 2019:4401–10.
[24] Zhu JY, Park T, Isola P, Efros AA. Unpaired image-to-image translation using cycle-consistent adversarial networks. Proc. ICCV (Venice, Italy) 2017:2223–32.
[25] Hinz T, Fisher M, Wang O, Wermter S. Improved techniques for training single-image gans *Proc*. CVF; 2021. p. 1300–9.
[26] Härkönen E, Hertzmann A, Lehtinen J, Paris S. GANSpace: discovering interpretable GAN controls. Proc. NIPS (virtual) 2020.