Multilayer Map Generation Using Attribute Loss Functions

Runze Tang
runze.tang@anu.edu.au
The Australian National University
Canberra, ACT, Australia

Penny Sweetser
penny.kyburz@anu.edu.au
The Australian National University
Canberra, ACT, Australia

ABSTRACT

Procedural Content Generation via Machine Learning (PCGML) has been studied to generate terrain maps, but many studies focus on height maps and lack human control. We propose a method based on Generative Adversarial Networks (GANs) to generate multilayer maps of terrain with statistical attributes as inputs to introduce more human control. Since the discriminators used in GANs are difficult to evaluate and lack transparency, we propose attribute loss functions, which work as a supervised approach to evaluate the statistical attributes of generated maps directly using differentiable functions for backpropagation. We tested combinations of two model architectures and different conditional normalisation methods and analysed their characteristics. We found that CGAN architecture with batch normalisation worked well in general, while SPADE block introduced more fragments, and channel-wise normalisation satisfied input conditions better but lost distribution diversity and inter-layer relationships.

CCS CONCEPTS

• **Computing methodologies** → *Computer vision tasks*; • **Applied computing** → *Computer games*.

KEYWORDS

procedural content generation via machine learning, generative adversarial networks, video games, terrain map generation

ACM Reference Format:
Runze Tang and Penny Sweetser. 2023. Multilayer Map Generation Using Attribute Loss Functions. In *Foundations of Digital Games 2023 (FDG 2023), April 12–14, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3582437.3587175

1 INTRODUCTION

Many recent video games include increasingly realistic terrain. However, creating realistic terrain or maps can be time-consuming. Modifying real-world maps, such as street or satellite maps, can present a time-saving approach. However, previous work in map generation often focuses on generating height maps only, while also lacking in human control [2, 12, 17]. In this work, we use real-world multilayer maps for our dataset to train a neural network to generate customised multilayer maps, including a height layer indicating the relative altitude and a segmentation layer indicating the ground categories, such as water and grassland. The generated maps maintain a rational relationship between layers, for example, water occurring in lower areas. We used Generative Adversarial Networks (GANs) [5] with conditions to generate maps. Introducing conditions can provide the game developer with more control over the generated maps. For the conditions, we used a vector of statistical attributes of the multilayer map.

The method by which a GAN discriminator measures the quality of an image is not explicit for humans to evaluate. In this work, we propose attribute loss functions that extract statistical attributes from the generated multilayer maps directly to evaluate whether they match the input conditions, such as average height in the height layer. The attribute loss functions work in a similar way to supervised learning: the statistical attributes are the labels and our attribute loss functions are customised differentiable loss functions.

We tested our attribute loss functions in combinations of model architectures and conditional normalisation methods. We found that CGAN with batch normalisation performed well in general. The element-wise normalisation method introduced more fragments because it focused more on each element, while the channel-wise normalisation method satisfied the conditions better but lost distribution diversity. We used the Normalized Relative Discriminative Score (NRDS) [20] to analyse the generated maps quantitatively and found that NRDS actually measures abnormal patterns rather than high-level structures or relationships between layers.

The primary contributions of our work are (1) using attribute loss functions to support GAN training explicitly and (2) generating multilayer maps controlled by input conditions. Our approach can also be extended to other tasks to support training. Further Procedural Content Generation (PCG) approaches can be used after generation to paint the texture and decorate the terrain with the generated map as a draft or guidance.

2 RELATED WORK

Procedural Content Generation (PCG) in video games is the algorithmic generation of game content [8]. PCG methods to generate maps include Perlin noise [15] and diamond square [4]. However, the generated maps can appear somewhat unnatural and unappealing [12]. In PCG via machine learning (PCGML), previous research on terrain generation uses real-world resources [2, 12, 17], such as height maps from NASA. Beckham and Pal [2] first generated height maps using GANs and then inferred the terrain texture of the generated height map. Spick et al. [17] produced a model that generates height maps that have similar patterns to the input regions, with details that vary extensively. Nunes et al. [12] compared several variants of GAN-based networks to generate height maps. To generate content with human control, Conditional GAN (CGAN) [11] simply concatenate initial noise and the conditions as the input

FDG 2023, April 12–14, 2023, Lisbon, Portugal

ACM ISBN 978-1-4503-9855-8/23/04.
https://doi.org/10.1145/3582437.3587175

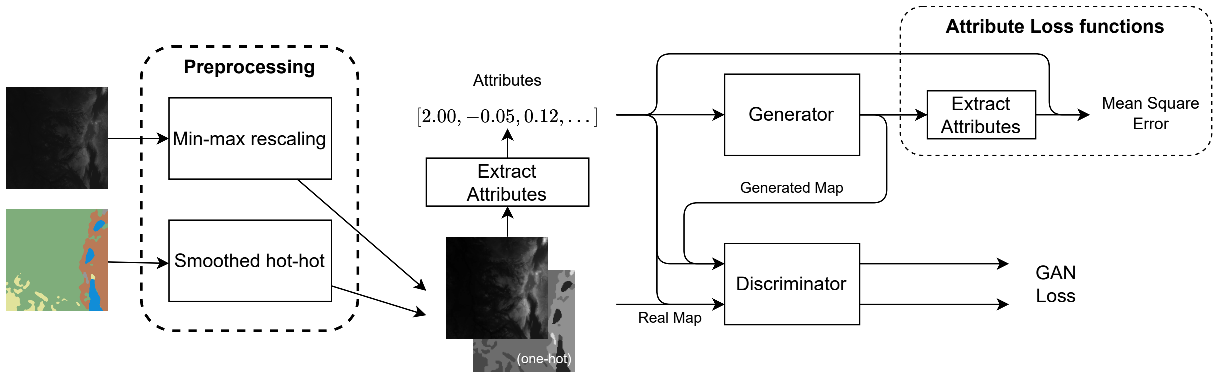

Figure 1: Overall workflow: preprocess the dataset, extract attributes, then train the model with the attribute loss functions.

to the GAN. GauGAN [14] inputs the condition at each normalisation layer (SPADE block) to prevent the deep model from forgetting the condition. In this work, our model generates multilayer maps, with rational inter-layer relationships and conditions that humans can control.

3 APPROACH

Our work aimed to generate multilayer maps with statistical attributes as input conditions so that the generation is under human control. The generated map contains a height map showing the relative height in an area and a segmentation map indicating the terrain category in different colours for each pixel. The overall workflow is shown in Figure 1.

3.1 Dataset

We used the dataset created by Pappas [13], in which each of the 5000 samples contains a height map and a segmentation map. The resolution of each map is 512×512 pixels and was resized to 128×128 in this work. Each pixel covers about $400m \times 400m$ of land. Height maps are single-channel images that encode the altitude information with 0 being sea level. Segmentation maps are coloured maps that indicate the terrain category for each pixel, where there are 7 categories in total represented by different colours. The range of most of the height maps only covers about half the range of 0 to 1. Therefore, min-max scaling is applied to map the altitude range in each height map from 0 to 1. The information loss of the altitude is recovered by using the weight and bias of height map min-max scaling as the attributes in training. We use smoothed one-hot encoding [7, 18, 19] on the segmentation maps.

3.2 Attribute Loss Function

For each sample, we chose 11 attributes to extract. Four attributes are for the height map: the weight and bias when applying min-max scaling and the mean and variance of height after min-max scaling. Seven attributes are for the segmentation map: the percentage of the area taken by each of the 7 ground labels.

The attribute loss functions first extract the attributes from the generated image and then calculate mean square error (MSE) for backpropagation. In our attribute loss function, we only consider the last 9 attributes, as the weight and bias when applying min-max scaling cannot be evaluated explicitly. For the other 2 attributes of the height map, the calculation of mean and variance is differentiable. For the attributes of the segmentation map, the percentage of the area taken by each ground label should be calculated in a differentiable way. Since the size of the image is fixed, calculating the proportion of each ground label is essentially counting pixels. We used smoothed one-hot encoding, so that before summation an amplified sigmoid function $sigmoid(\alpha(x - 0.5))$ is applied, where $x \in [0, 1]^{128\times128}$ is a one-hot encoded category of the segmentation map. Amplification factor α is used to polarise the value in the map, which is a hyperparameter. We chose $\alpha = 20$.

3.3 Models

We tested three approaches to providing conditions to the model: conditional GAN (CGAN), element-wise normalisation (SPADE block), and channel-wise normalisation (CWN). In CGAN, the conditions are only input at the beginning, while the SPADE block and CWN repeat the condition in the normalisation layers to the model multiple times.

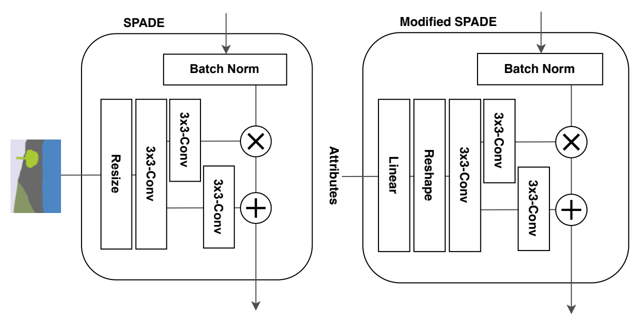

Figure 2: Original SPADE block (left) and our modified version (right).

In CGAN [11], the conditions are concatenated with the noise as the input to the model, which is suitable for models that are not too deep, otherwise the model may forget the conditions quickly. To solve the problem of forgetting the conditions, GauGAN [14] used SPADE blocks, an element-wise normalisation, to repeat the conditions to the network. An element here means a value of a pixel in a channel. The original SPADE block takes the segmentation map as input and informs each pixel to which object it belongs, as Figure 2 shows. However, in our work, the condition is a vector (attributes) rather than a matrix (segmentation map). Since the idea of SPADE is to generate independent weight and bias of normalisation based on the input for each element, we applied a modification. As Figure 2 shows, we used a linear layer followed by reshaping to replace the resize operation in the original SPADE block. The linear layer and reshape were used to generate a map that has the same shape as the input image, as did the resize operation in the original SPADE block. In our work, attributes correspond with map layers, therefore, we tested a channel-wise normalisation, which calculates weight and bias for normalising each channel with attributes as input.

4 EVALUATION

We tested different combinations of model architectures, CGAN [11] and GauGAN [14], while the number of parameters is reduced, with different normalisation methods: batch normalisation (BN) [10], modified SPADE block, and CWN. We also tested different GAN loss functions, including the original version [5], the logD alternative [1], and WGAN-GP [6]. In this section, we report on our performance evaluation by comparing the results via NRDS and observation.

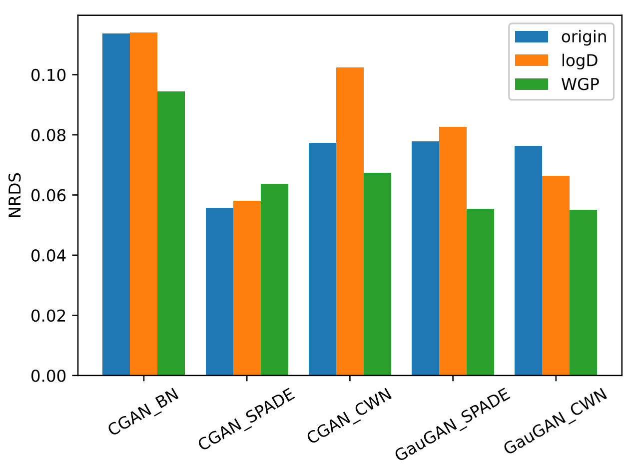

Figure 3: NRDS for each combination of models, normalisation methods, and loss functions.

4.1 Quantitative Analysis (NRDS)

Most of the evaluation metrics for image generation [3] rely on a label-specific pre-trained classifier, such as Inception score [16] and Fréchet Inception Distance (FID) [9]. In this work, we used the Normalised Relative Discriminative Score (NRDS) [20]. NRDS is the area under the epoch-label curve of a randomly initialised classifier trained to classify whether the input image is real or generated. The closer a generated map is to the real one, the more epochs the classifier will take to learn to distinguish between them and the larger the area under the epoch-label curve. Therefore, higher NRDS means the generated content is more similar to the real content according to the classifier.

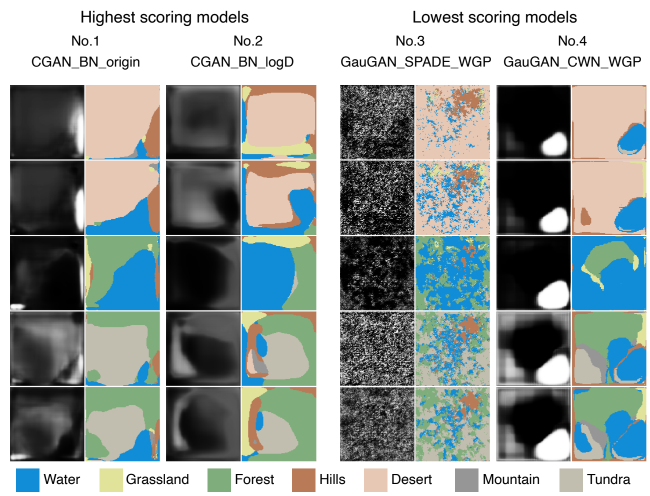

Figure 4: Samples generated by the two models with the highest NRDS (left), and two lowest (right). The grayscale images are height maps and the coloured images are segmentation maps. Each row contains samples with the same attributes.

As shown in Figure 3, the CGAN architecture with BN achieved the highest score. Generally, models with the SPADE block achieved a lower score than others. As the underlying principle of NRDS depends on the features the classifier learned, we combine NRDS with qualitative analysis (Figure 4) to discuss what the score is measuring. Comparing these samples visually, we can infer that, rather than high-level features, such as the relationship between map layers or the distribution of ground categories, the NRDS evaluates obvious artefacts, such as fragments and white ellipses.

4.2 Qualitative Analysis

To visualise the generated maps, we divided the generated maps into three types by their appearance, as shown in Figure 5. For the height maps, both Type 1 and 2 are too smooth compared with the real one, and Type 3 height maps are too fragmented. For the segmentation map, Type 1 maps have few details and labels; Type 2 maps have more details and stronger label diversity but are still too smooth and lack erosion patterns; Type 3 maps are too fragmented. Among all three types of maps, only Type 2 has obvious inter-layer relationships: water occurs in low-lying places and is blocked by hills or mountains in higher places. Type 1 maps were generated by CGAN architecture with CWN, which suggests that the CWN worked, but over-emphasised the attribute and loses the richness of ground categories and the relationship between the height map and segmentation map. Type 2 maps were generated by the CGAN model with BN and GauGAN model with CWN. We can infer that the complexity of GauGAN cancelled out the tendency of CWN to simplify maps. All models that use SPADE blocks generated maps of Type 3. We also found that CGAN with SPADE blocks

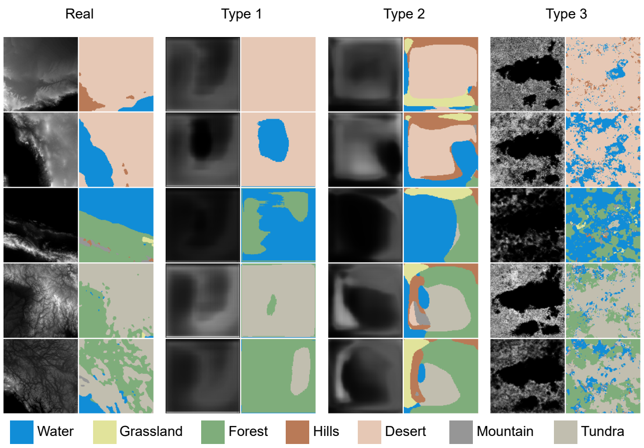

Figure 5: Three main types of the generated maps and the corresponding real maps. The grayscale images are height maps and the coloured maps are segmentation maps. Each row contains samples with the same attributes.

tended to generate fewer fragments than the GauGAN model with SPADE blocks, as the original GauGAN model is more complex and designed to use the SPADE block. The reason why the SPADE block causes fragments is that the element-wise operation pays more attention to each element itself rather than the relationship between neighbouring elements.

5 DISCUSSION AND CONCLUSION

Together with attribute loss functions, we trained multiple models, each of them with a combination of basic architecture, normalisation methods, and loss functions of GAN. We evaluated the generated maps by both observings and using NRDS. We found that the CGAN architecture can generate relatively good maps compared with GauGAN. Compared with CGAN, GauGAN has a far more complex architecture and more parameters to train, and takes fewer epochs to achieve its best result. Therefore, we can conclude that in this work, GauGAN learned faster but also exhibited more overfitting since the dataset with 5000 samples is too small. Therefore, data-augmentation techniques will be considered in future work. Through qualitative analysis, we found that element-wise normalisation (SPADE blocks) introduced more fragments that may suit some special map regions, such as a marsh. As the original task that the SPADE block was proposed for was generating real images, the value change of neighbour pixels in real images is larger than the maps in this work. Channel-wise normalisation should be suitable for this task because there are attributes that correspond to each channel. However, in practice, it loses the distribution richness of categories and the inter-layer relationship.

This work aimed to generate multilayer maps preserving meaningful relationships between layers with statistical attributes as input conditions to introduce more human control. Similar to supervised learning, we proposed attribute loss functions to work directly on the output of the generator of the GAN to evaluate the degree to which the generated map satisfied the input attributes explicitly for humans to understand while differentiable for the model to backpropagate. Our idea of using attribute loss functions to apply a differentiable supervised learning technique directly to the generated content could be a powerful way to enhance the quality of the generated content given specific conditions. We categorised the generated maps into three types and analysed their characteristics: element-wise normalisation (SPADE block) introduces more fragments; maps generated by models using CWN satisfy the input attributes better than others but lost distribution diversity of ground categories; CGAN architecture with batch normalisation generated maps well in general. In quantitative analysis, we found that NRDS actually measured the abnormal patterns in the generated maps rather than high-level features. For future works, data-augmentation techniques should be considered and how the model works on abnormal attributes should be examined.

REFERENCES

[1] Martin Arjovsky and Léon Bottou. 2017. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862* (2017).
[2] Christopher Beckham and Christopher Pal. 2017. A step towards procedural terrain generation with gans. *arXiv preprint arXiv:1707.03383* (2017).
[3] Ali Borji. 2019. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding* 179 (2019), 41–65.
[4] Alain Fournier, Don Fussell, and Loren Carpenter. 1982. Computer rendering of stochastic models. *Commun. ACM* 25, 6 (1982), 371–384.
[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).
[6] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. *Advances in neural information processing systems* 30 (2017).
[7] Tamir Hazan, George Papandreou, and Daniel Tarlow. 2016. *Perturbations, Optimization, and Statistics.* MIT Press.
[8] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. 2013. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 9, 1 (2013), 1–22.
[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* 30 (2017).
[10] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. PMLR, 448–456.
[11] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).
[12] Vasco Nunes, João Dias, and Pedro A Santos. 2023. GAN-Based Content Generation of Maps for Strategy Games. *arXiv preprint arXiv:2301.02874* (2023).
[13] Thomas Pappas. 2020. Earth Terrain, Height, and Segmentation Map Images. https://www.kaggle.com/datasets/tpapp157/earth-terrain-height-and-segmentation-map-images
[14] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2337–2346.
[15] Ken Perlin. 1985. An image synthesizer. *ACM Siggraph Computer Graphics* 19, 3 (1985), 287–296.
[16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. *Advances in neural information processing systems* 29 (2016).
[17] Ryan J Spick, Peter Cowling, and James Alfred Walker. 2019. Procedural generation using spatial GANs for region-specific learning of elevation data. In *2019 IEEE Conference on Games (CoG)*. IEEE, 1–8.
[18] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.
[19] Jiongmin Zhang, Ke Jia, Jinmeng Jia, and Ying Qian. 2018. An improved approach to infer protein-protein interaction based on a hierarchical vector space model. *BMC bioinformatics* 19, 1 (2018), 1–14.
[20] Zhifei Zhang, Yang Song, and Hairong Qi. 2018. Decoupled learning for conditional adversarial networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 700–708.