Dual Critic Conditional Wasserstein GAN for Height-Map Generation

Nuno Ramos
Instituto Superior Técnico
University of Lisbon
Lisbon, Portugal
nuno.m.ramos@tecnico.ulisboa.pt

Pedro A. Santos
Instituto Superior Técnico / INESC-ID
Lisbon, Portugal
pedro.santos@tecnico.ulisboa.pt

João Dias
University of Algarve *and* CCMAR
and INESC-ID
Faro, Portugal
jmdias@ualg.pt

ABSTRACT

Traditionally, video-game maps are either made by hand, requiring many man-hours to produce good results, or made using Procedural Content Generation (PCG) techniques, which rely on a predetermined algorithm to generate every feature of the map. More recent studies have tried an approach using Deep Learning algorithms, which have their own limitations, in particular taking away the creative freedom of the designers. To circumvent this problem we propose a system that transforms low fidelity sketches into realistic height-maps through a Deep Learning model we call the Dual Critic Conditional Wasserstein GAN (DCCWGAN), thus providing high visual quality without removing control from the user. The presented system is capable of producing images that resemble the received input, and a user study with 79 participants showed that observers are not able to distinguish between earth-based height-map images and the images generated by our system.

CCS CONCEPTS

• **Computing methodologies** → *Neural networks.*

KEYWORDS

Height-map, Deep Learning, Image-to-Image Translation, GAN, Conditional GAN

ACM Reference Format:
Nuno Ramos, Pedro A. Santos, and João Dias. 2023. Dual Critic Conditional Wasserstein GAN for Height-Map Generation. In *Foundations of Digital Games 2023 (FDG 2023), April 12–14, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3582437.3587183

1 INTRODUCTION

Maps are a crucial part of most video-games: in exploration games different areas can provide interesting challenges for the player, in strategy games more defensible geography can be considered a critical asset. If a video-game contains a map it is a reasonable assumption that the quality of the player experience is somewhat tied to the quality of this map.

In the video-game industry maps are made in one of two ways: either by hand, which is a very time-consuming task, whose results depend both on the prowess of the designer, who designs the challenges present in the map, and the artist, whose job is to implement those ideas; the other method is to procedurally generate those maps, which involves designing an algorithm to create a map from a set of parameters and random values; this approach has it's own problems, namely that it's results have sub-par quality and take control away from the designer, who, for the most part can no longer specify exactly which geographic formation appears where. Recent works have tried to overcome the limitations of traditional PCG techniques by implementing machine learning algorithms, using real-world geographic information to train networks created for this specific purpose[5, 7, 10]. While this approach produces promising results it has a fundamental flaw: similarly to PCG techniques, it removes control from the designer.

Another line of works [2, 8] uses conditional GANs to introduce some measure of designer control to the result, which became possible due to recent advances in the area of image-to-image translation [3, 4, 6, 9]. In this line of research, we propose a conditional WGAN that uses two critics - one focused in evaluating quality or realism of an image, and another focused in evaluating how close the generated image resembles a rough sketch of a map - to alternately train the generator. Using this model we developed a tool that allows developers to draw a rough sketch of a map that will be transformed into a realistic version of the terrain depicted in the sketch, thus maintaining the visual quality while reducing effort without having to sacrifice control.

2 SYSTEM OVERVIEW

In [5] the authors focused heavily on testing multiple Deep Learning architectures for generating height maps and comparing their results. We adapted one of the most successful models described, the Wasserstein GAN [1], iterating the model based on results achieved.

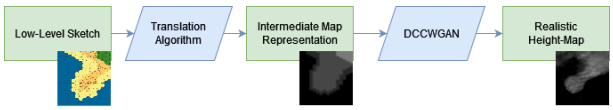

Figure 1: Data flow for the system, user creates low-level sketches (left) and the system outputs Realistic Height-maps (right).

The data flow for our system (illustrated on Figure 1) begins with the low-level sketch, which is created by the user and exists only during inference. While we illustrate a specific type of sketch it is important to stress that the system is very easily adapted to use a different tool to create sketches. The low-level sketch is then used as input to a translation algorithm, which transforms it into what we call the Intermediate Map Representation (IMR) format.

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

ACM ISBN 978-1-4503-9855-8/23/04.
https://doi.org/10.1145/3582437.3587183

For our system to function there needs to be a format that will be used as input by the user, but is also able to be created from the ground-truth data set, so as to compare if the content matches during training, this format is the Intermediate Map Representation (IMR). The IMR format exposes the average height of each cell in a hexagonal grid. The IMR outputted by the translation algorithm is then used as input to our deep learning model, the Dual Critic Conditional Wasserstein GAN (DCCWGAN), both during inference and training, resulting in the height-map.

2.1 Dual Critic Conditional WGAN

In order to choose an architecture for our deep learning model we started by dividing the problem into two distinct learning processes:

- Create realistic maps.
- Create maps whose content corresponds to the given input.

Given this division, we opted for an architecture with a Generator, but two distinct critics, one for each of the necessary learning processes. The first of these Critics evaluates only the visual qualities of the images generated, and forces the Generator to create more and more realistic images; for this reason we call this the Realism Critic. The second critic is responsible for evaluating how well the contents of the generated maps correspond to the input used in their generation; this critic is called the Conversion Critic. Training is done using one Critic at a time, depending on the current loss. The decision of which Critic to use is explained in further detail in Section 2.6.

2.2 Generator

The Generator is the network responsible for generating images. It accepts two separate inputs: the IMR and a noise vector. The IMR controls the general layout of the final result, while the noise vector, similarly to other GAN's, provides a source of entropy, adding a layer of randomness to the output and allowing the same IMR input to generate multiple different results.

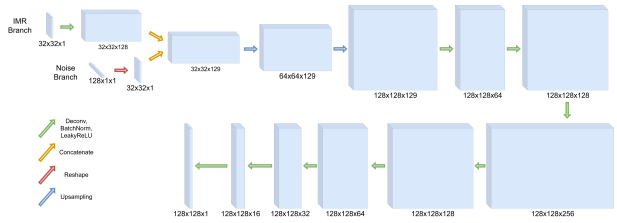

Figure 2: Structure of the Generator.

Figure 2 represents the complete structure of the Generator. All deconvolutional layers use kernel size of 3 and stride of 1, except for the layer following the IMR input, which uses a kernel size of 3 and dilation rate of 2, and the final convolutional layer, which uses a filter size of 5.

2.3 Conversion Critic

The Conversion Critic, unlike a traditional WGAN critic, receives a paired input: a real or fake image and an IMR. The IMR given is either generated from the real data set or, in the case of fake images, the IMR used as input for the Generator. The purpose of this network is to discriminate whether the general content of the image matches that of the IMR given. In the case of real images the pair should easily match since the IMR is created from the image itself; in the case of fake images the network attempts to correct the generator to force this content matching.

Figure 3 depicts the structure of the Conversion Critic, illustrating the two separate inputs (IMR and full image), which are concatenated channel-wise. The goal of this network is to learn to give a good score to examples where the IMR and the higher-resolution images are structurally similar, and a bad score otherwise. All convolutional layers use a filter size of 4, except for the layer following the IMR input, which uses a kernel size of 3 and dilation rate of 2, and the final convolutional layer, which uses a filter size of 5. When image width and height decreases this is done using a convolutional layer of stride 2.

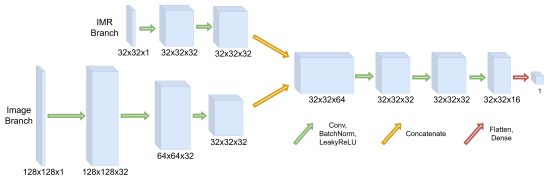

Figure 3: Structure of the Conversion Critic.

The Conversion Critic has a much simpler task than that of the Realism Critic, therefore requiring less parameters and less total memory to train. It is worth noting that the Image Branch of this network downsamples very rapidly, this is because, due to the way in which we defined the IMR format, it is much easier to confirm that the two branches have similar geographical content at a lower resolution.

2.4 Realism Critic

This critic functions exactly as a critic in any generic WGAN. It takes as input either a real or fake image and outputs a score of the realness of the image received. Unlike the content critic, this critic does not receive the IMR, and therefore judges only the visual quality of the image, and not its accuracy.

Figure 4: Structure of the Realism Critic.

Figure 4 represents the complete structure of the Realism Critic. All convolutional layers in this critic use a stride value of 1 and filter size of 4. While on a surface level this network appears smaller, as it contains less layers, it contains over 7 times more parameters. This is because while the Conversion Critic only guides the broader strokes of the image, the Realism Critic is responsible for the finer detail, which is a more difficult task that will be held to a higher standard by the end user.

2.5 The data sets

As a starting point we were graciously given the data set used by the researchers of GAN-Based Content Generation of Maps for Strategy Games [5], which is itself a filtered and augmented version of the Shuttle Radar Topography Missions (SRTM) [1] data set, created by NASA. This data set is used as the ground-truth for this research, but it requires a pair to be used as input of the model.

The pair to the ground-truth data set is the Intermediate Map Representation (IMR), and is used both in training the Deep Learning model and is what the user will create as input to the final system. We used a simple grid of discrete values in the range [0,1], corresponding to the average elevation value in that area. The number of levels used was five, but the same principle works for any number of discrete values.

To create the IMR format, we analyze each image in the ground-truth data set and obtain the average elevation of all pixels that correspond to each hex cell. Each of these averages is then compared to the list of possible values for the IMR format and is attributed one of the two closest values randomly based on how close it is to either.

2.6 Training

Algorithm 1: WGAN training algorithm with $n\_critic = 2$ and $batch\_size = 64$

```
Normalize p_data between −1 and 1;
for epochs do
    shuffle p_data;
    half_batch = batch_size / 2;
    iterations = number of images of p_data / half_batch;
    for iterations do
        Choose Critic C to train according to loss;
        for n_critic do
            z_vectors = half_batch samples from N(μ, σ²);
            real_images = half_batch images from p_data;
            fake_images = half_batch samples from
              G(z_vectors);
            y_real = half_batch size vector of value −1;
            y_fake = half_batch size vector of value 1;
            Train C with real_images labelled as y_real
              using gradient descent with Wasserstein
              estimate;
            Train C with fake_images labelled as y_fake
              using gradient descent with Wasserstein
              estimate;
        z_vectors = batch_size samples from N(μ, σ²);
        y_gen = half_batch size vector of value −1;
        Train G with z_vectors and y_gen labels using
          gradient descent with Wasserstein estimate;
```

The training of our system, for each GAN, is identical to the algorithm used to train any generic WGAN, as presented in Algorithm 1, the only difference from this algorithm is that our system uses not one, but two separate GAN's that share the same Generator, and as such, it is necessary to decide which GAN (or Critic) to train at any given point, which is done depending on the critics losses'. After some empirical tests, we realized that the Conversion Critic was easier to train (due to having a simpler task and a simpler network) and thus does not need to train as often. The final criterium used is the following: if the loss of the Conversion Critic in fake examples is more than twice the size of the Realism Critic in fake examples, then the Conversion Critic will be selected, otherwise select the Realism Critic.

[1] https://www2.jpl.nasa.gov/srtm/

3 RESULTS

All tests were done on a server using a Intel(R) Xeon(R) W-2223 CPU with 98 GB's of RAM and two GeForce RTX 3090 GPU's, however, only one was ever used at a time.

All the experiments described are compiled with the RMSProp optimizer, with a learning rate of 0.00025 for the Conversion GAN and 0.0005 for the Realism GAN. The training was done for 5000 epochs.

3.1 Conversion Results

In this section we present the results obtained by our system in terms of being able to convert a low-resolution image or sketch into an image with higher quality or detail.

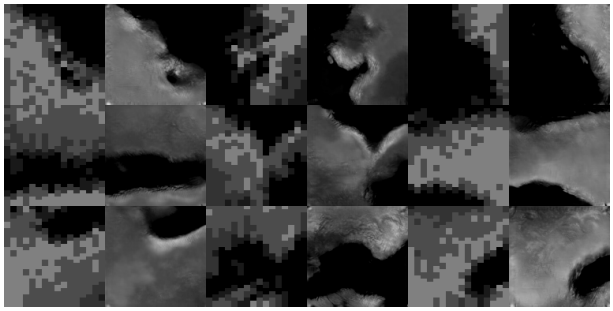

Figure 5: Examples of pairs of images. On the left side of the pair we have the lower-resolution IMR and on the right side the final image generated by our system.

In Figure 5 we can observe some maps generated from our final model. We believe these results to be of high quality, reproducing the given content successfully, with few artifacts or added noise. We can observe that the model maintains the general profile of the terrain while adding some texture. It should be noted that the contour of the coastline is kept identical to the IMR used, while the elevation within the land differs slightly, fluctuating depending on the noise vector given. This vector requires a careful balancing of the model as, if the vector is not given enough importance, then there will be little randomness in the images, in the extreme case each IMR input can only generate a single output map. On the other hand, if the noise vector is given too much importance then the IMR input will not be respected and the output map will not represent the given input.

In order to analyse how well the system creates any type of terrain from a rough sketch, and to ensure it wasn't overfitting, we

created a new set of examples, by first creating 32 hexagonal maps using a hexagonal map creation tool, and subsequently translating the hexagonal maps into the IMR format.

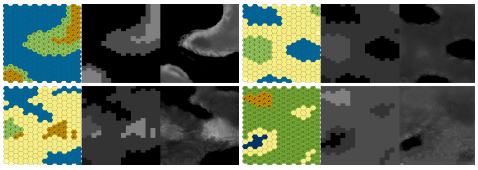

Figure 6: Examples of maps generated from the custom IMR data set. For each set: Left: Original hex map. Center: IMR. Right: Resulting height-map.

Figure 6 represents some of the results obtained from the model when given IMR representations not present in the training data. These examples could be problematic to our model, not only because they were never observed, but also because they can express terrain that may be unrealistic. Although the quality of the generated images for these examples is sometimes lower, by generating multiple samples of the same input, some have quality on par with the previously observed results, further demonstrating the importance of the noise vector.

3.2 Realism Results

In order to evaluate our results on the goal of generating realistic images we wanted to determine if users were unable to distinguish images generated from our system and images from the original NASA STRM data set. To that end we conducted a user test where participants were shown 20 height-maps and corresponding 3D-renders, 10 of which from the NASA SRTM data set, and 10 generated by our system. Participants were then asked to evaluate the origin of each map, using the sentence “This map represents geographic information from the Earth" and asking participants how much they agree with the sentence, in a Likert scale of 1 to 7.

We obtained a total of 79 participants in the study. Figure 7 shows a boxplot with the values obtained for both groups of images. Because our data is non-parametric, we used the Wilcoxon’s paired rank test, from which we determined that there was no statistical difference between the two populations, in other words, participants were unable to distinguish between ground-truth maps and maps generated using our system ($Z = -0.399, p = 0.69$).

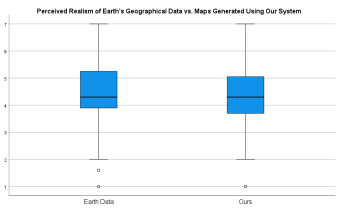

Figure 7: User perception of realism of maps generated with Earth’s geographical data, vs. generated by our system.

4 CONCLUSION

Our goal was to provide an alternative way of generating height-maps for video-games. We needed a system that would create realistic and visually appealing results, without requiring too much work from the part of user, but still allowing the user to specify desired geographical features. In order to accomplish this goal we introduced the Dual Critic Conditional Wasserstein GAN (DCCWGAN), a new type of conditional WGAN using two critics: The first Critic to evaluate the content of the input matches that of the generated map, while the second Critic guides the results to be more realistic. Results show the content of the outputted maps closely match those of the supplied input, and that in terms of realism a test with 79 human participants showed that observers are not able to distinguish between real images and images generated by our system. Overall, we consider that, while there is room for improvement, we achieved the goals we set out for, and contributed to existing knowledge by implementing a system that performs a form of sketch-to-image translation using multiple critics.

Acknowledgments

This work was supported by national funds through FCT, Fundação para a Ciência e a Tecnologia, under projects UIDB/04326/2020, UIDB/50021/2020, UIDP/04326/2020 and LA/P/0101/2020.

REFERENCES

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein GAN. arXiv:1701.07875

[2] Éric Guérin, Julie Digne, Éric Galin, Adrien Peytavie, Christian Wolf, Bedrich Benes, and Benoît Martinez. 2017. Interactive Example-Based Terrain Authoring with Conditional Generative Adversarial Networks. *ACM Trans. Graph.* 36, 6, Article 228 (nov 2017), 13 pages. https://doi.org/10.1145/3130800.3130804

[3] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and Improving the Image Quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

[4] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral Normalization for Generative Adversarial Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.* OpenReview.net.

[5] Vasco Nunes, João Dias, and Pedro Santos. 2022. GAN-Based Content Generation of Maps for Strategy Games. In *GAME-ON 2022.*

[6] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. GauGAN: Semantic Image Synthesis with Spatially Adaptive Normalization. In *ACM SIGGRAPH 2019 Real-Time Live!* (Los Angeles, California) *(SIGGRAPH ’19).* Association for Computing Machinery, New York, NY, USA, Article 2, 1 pages.

[7] 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)* (London, United Kingdom). IEEE Press, 1–8. https://doi.org/10.1109/CIG.2019.8848120

[8] Georgios Voulgaris, Ioannis Mademlis, and Ioannis Pitas. 2021. Procedural Terrain Generation Using Generative Adversarial Networks. *2021 29th European Signal Processing Conference (EUSIPCO)*, 686–690.

[9] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-Resolution Image Synthesis and Semantic Manipulation With Conditional GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[10] Andreas Wulff-Jensen, Niclas Nerup Rant, Tobias Nordvig Møller, and Jonas Aksel Billeskov. 2018. Deep Convolutional Generative Adversarial Network for Procedural 3D Landscape Generation Based on DEM. In *Interactivity, Game Creation, Design, Learning, and Innovation*, Anthony L. Brooks, Eva Brooks, and Nikolas Vidakis (Eds.). Springer International Publishing, Cham, 85–94.