IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 29, 2020, accepted December 15, 2020, date of publication December 29, 2020,
date of current version January 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3048083

Generation of Synthetic Elevation Models and Realistic Surface Images of River Deltas and Coastal Terrains Using cGANs

LUIS OSWALDO VALENCIA-ROSADO, ZOBEIDA J. GUZMAN-ZAVALETA, (Member, IEEE), AND OLEG STAROSTENKO

Department of Computing, Electronics and Mechatronics, Universidad de las Américas Puebla (UDLAP), Cholula 72810, Mexico

Corresponding author: Oleg Starostenko (oleg.starostenkoudlap.mx)

This work was supported in part by the Mexican National Council of Science and Technology (CONACYT) scholarships.

ABSTRACT Terrain generation aims to automatize the procedure of creating landscapes using a computer system. The generation models must follow different terrains' topographical features, such as areas with river deltas and other regions where water bodies affect the natural landscapes. It is possible to generate more realistic terrains thanks to improvements in computer graphics techniques and deep learning models that use specific hardware. However, the advance on the generation of terrains that include river deltas, fjords, and waterfalls has not had the same pace as other more studied landscapes. Therefore, as a contribution to the advance of the research of terrain generation with water bodies using generative models, this paper presents the DRCA2020 dataset, which is useful for supervised training. The proposed dataset contains eight different types of real-world satellite images. These images are grouped by the same geographical location. There are 13,184 groups; each one has three RGB surface images, a water coverage map, three binarizations of water coverage, and a digital elevation model (DEM). Additionally, this paper proposes the use of a cGAN composite model, trained with the DRCA2020 dataset, for generating synthetic DEMs from water coverage images and therefore, to create realistic surface texture images with promising validation results.

INDEX TERMS cGANs, generative model, river delta, river delta dataset, satellite imagery, terrain generation.

I. INTRODUCTION

Procedural terrain generation (PTG) is the process to semi- or automatically create virtual terrains with computer systems. Terrain generation is employed in several fields, including virtual reality [1], video game production [2], and in Machine Learning for augmentation of training sets. Moreover, PTG has become more prominent in the last decade as video games have raised their production costs [3]. According to its definition in geomorphology, a terrain consists of different land features such as mountains, valleys, canyons, rivers, lakes, fjords, waterfalls, river deltas, coastlines, among others. Natural processes such as erosion and deposition mold these features. Erosion is when terrain material is removed and carried away by wind or water. In counterpart, deposition is when some material is accumulated in another area and forms new landscape. Additionally, there are faster natural changes provoked by avalanches, fires, storms, floods, or earthquakes. Moreover, human activities generate changes in terrains [4]. Therefore, one of the challenges of the PTG is to recreate different types of terrain features more realistically, especially in complex landscapes, such as the ones including features like river deltas, fjords, and waterfalls.

Emerging approaches, such as the presented in [5] and [6], generate river deltas using stochastic or Machine Learning methods respectively, with promising results. In counterpart, Generative Adversarial Networks (GANs) have more impressive results in creating digital data such as images, audio, and videos. GANs are a class of Machine Learning techniques that consists of two trained models, a generator and a discriminator, that simultaneously improve the realistic data generation [7]. As other Machine Learning techniques, GANs require data to train the models to recognize the main features that describe it. In the case of terrain generation, it is preferred to use real-world data from satellite imagery. Satellite imagery includes height maps, land surface images, sunlight

The associate editor coordinating the review of this manuscript and approving it for publication was Songwen Pei.

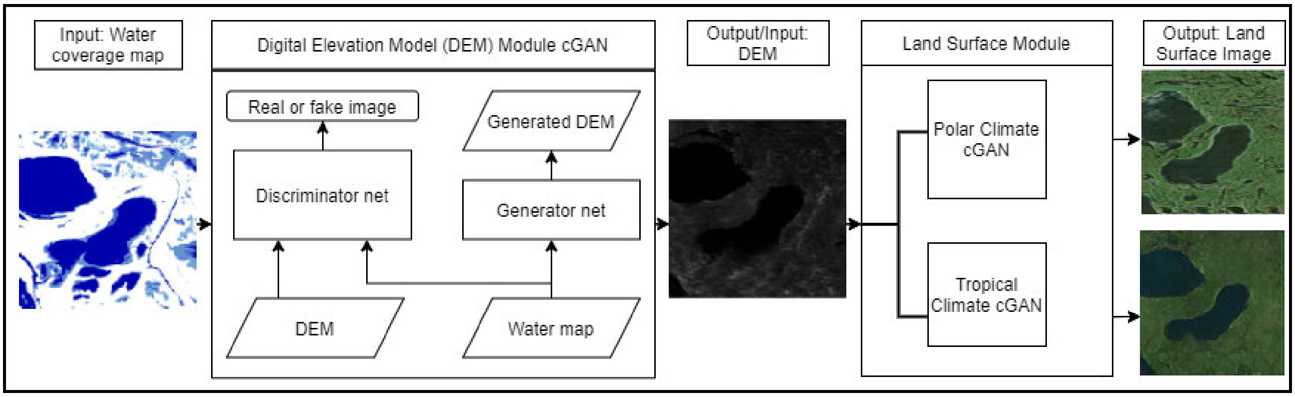


FIGURE 1. The pipeline of the proposed methodology. A Water Coverage Map is required to create a Digital Elevation Model (DEM), which is the G(x) output from the cGAN module. Then, this DEM is the input for another cGAN module to generate a land surface image according to the selected climate.

exposition maps, and humidity maps [8]. These are publicly available from institutions such as the National Space Agency (NASA) through its *EarthData* portal [9], the United States Geological Survey (USGS) in its *EarthExplorer* platform [8], and from the European Space Agency (ESA) in the portal of the *Copernicus Open Access Hub* [10].

Additionally, some satellite imagery datasets are available and ready to be used for adjusting object recognition models in urban environments, *e.g.* those presented in [11], [12], and [13]. Other datasets provide semantic segmentation of images in cities or towns, as in [14], [15], and [16] for instance. Some others are used in land coverage classification, *e.g.* [17] and [18]. A further description of these datasets is found in Section III. However, there is a lack of public datasets that include processed images for river deltas and coastal terrain generation, which also delays the contributions with practical models to create landscapes with water bodies. Therefore, there is a need for specific datasets for understanding and extracting the main features of terrains with river deltas and coastal areas, for their automatic generation, as well as approaches that focus on creating those kinds of landscapes.

Therefore, this paper's primary purposes are to present an approach for terrain generation focused on areas with river deltas and coastal regions and to introduce a specialized dataset for training and assessment purposes. On the one hand, this approach uses conditional Generative Adversarial Networks (cGANs) on a two-stage basis: the first one is to generate an architecture for creating synthetic Digital Elevation Models (DEMs) and the second one to generate the terrain textures for the realistic surface images (see Fig. 1). On the other hand, this paper introduces the Deltas, Rivers, and Coastal Areas 2020 - DRCA2020 dataset which collects satellite images with the mentioned water body areas. The purpose of DRCA2020 is to provide real-world imagery to improve the naturalness of generated terrains in different applications such as in simulation, game design, and hydrology research. The proposed approach is focused on the generation of terrains with two opposite and distinctive climates: tropical and polar. Both climates are included in the DRCA2020 dataset. Another advantage of the proposed approach is that it can accept user-drawn water coverage maps to control the delta generation.

The organization of this paper is as follows. Section II presents a brief analysis of terrain generation techniques, including a review of related works with river delta generation. Then, Section III describes the proposed DRCA2020 dataset. Section IV shows the methodology for the terrain generation and the cGANs architectures. Section V presents the setup for the cGANs followed by Section VI with the presentation and discussion of the experimental results. Finally, the concluding remarks are in Section VII.

II. RELATED WORK

The pace of automatic and realistic terrain generation focused on specific water bodies is different from the terrain creation in general. There are a few approaches with potential but partial results. For instance, Teoh [5] proposed a method for generating coastal terrain features. That work introduces a stochastic approach that generates a river delta with a single branching point. The generation process of the delta in [5] is as follows: first, it generates a river that reaches the sea; second, it creates a semicircle of new land around the river mouth and randomly selects points on the updated coast; finally, the former river mouth is joint to these points with distributary channels. The main disadvantage of that generation process is that it is not generalized to other types of deltas. Another example of river delta generation, proposed by Nesvold [6], uses a Wasserstein GAN trained with 20,000 multispectral satellite images which are subsections of 40 different river deltas. These subsections have different scales to capture the changes in geometrical shape features of the deltas. However, the goal of Nesvold's work is to learn the depositional patterns of the river deltas. Similarly, the method presented by Seybold [19] is a simulation of the water flow and the erosion and deposition processes. It can generate realistic deltas; however, the amount of data and needed processing renders is unsuitable for being the backbone in a procedural generation application. In general, geological simulations are significantly different from the methods that focus on graphic simulation. For instance, geological simulations require soil composition, terrain slope, water presence and rainfall volume. Moreover, geological methods may be restricted to a mathematical description of the river delta behavior with no graphical representation [20]. Additionally, other works can produce river networks but do not generate deltas, *e.g.* [21] and [22]. Other works focus in specific features such as waterfalls, *e.g.* [23].

Several researchers have used Generative Adversarial Networks (GANs) to tackle the terrain generation problem. For instance, Ping *et al.* [24] use GANs in conjunction with classic Convolutional Neural Networks (CNNs) for creating video game maps represented with height maps. A prominent work that combines the use of sketches and conditional GANs (cGANs) was presented by Guérin *et al.* [25]. That approach uses the Pix2Pix framework [26] to perform a translation between two images. Guérin *et al.* train the cGANs to generate height maps from sketch lines representing mountain ridges, river courses, and altitude cues. Another method that accept user-drawn inputs is presented by Zhou *et al.* [27]. In that seminal paper, the mountain ridges are identified from real-world height maps. Then, some patches are extracted and use it over user-generated sketches to create new mountain ridges. Other works do not need to have a mountain ridge to be explicitly drawn; instead, a sketch represents a low-definition height map as input, as it was proposed in [28]. Based on altitude differences, each part of the map is modeled after different terrain features. These methods generate mountain ranges, plateaus, or canyons, but they cannot create river deltas or coastal features. The evidence points that those excellent results for mountain landscapes can be transferred to the creation of terrains with river deltas by the use of a specialized training to fit similar models to this new scenario. Therefore, the proposal in this paper uses the well-known Pix2Pix framework with a single feature, the presence of water, to generate river deltas and coastal lowlands. The Pix2Pix model is fitted to this new scenario using the proposed DRCA2020 as the training dataset.

A. SPECIALIZED DATASETS

Many datasets use satellite imagery, which is ready for fitting Machine Learning models, but most of them are focused on urban areas; consequently, those are not suitable for natural landscape generation. Some of those datasets provide annotated images that can also be used for object detection tasks. These could be focused on just one type of object such as cars [11] or ships [12]; or in multiple types of objects including ships, planes, swimming pools, courts, helicopters and trucks [13]. Drone imagery datasets provide images similar to those of satellites, and provide annotations on smaller subjects like pedestrians, cyclists or skateboarders [29]. Other datasets are made up of images with annotations on areas instead of objects, there are useful for fitting semantic segmentation models. For example, the *RoadNet* dataset [14] is used for urban road identification. On the same problem, Isola [26] presented a dataset for creating road maps out of surface images using cGANs. The *xBD* dataset presented in [16] focuses on the aftermath of hurricane disasters. Finally, Maggiori presents a dataset for building detection [15].

In counterpart, there are datasets which are not focused on inhabited environments, like those used for cloud detection, *e.g.* [30] and [31]. Others provide annotations on soil composition [32] or land coverage [17], [18]. However, those public datasets do not provide enough images of river deltas and their coastal plains. For that reason, there is a need to provide a publicly available dataset which is focused on river deltas and coastal terrains to facilitate the research on the generation of these kind of terrains.

III. THE PROPOSED DRCA2020 DATASET

The DRCA2020 dataset contains satellite images from the principal river deltas of the world, which were selected based on the list provided by Coleman and Huh [33]. The dataset includes other water bodies for a total of 75 deltas and six bays, and it is available online at https://github.com/DRCA2020/Tropical-Rainforest-and-Monsoon. Hence, this dataset collects real-world imagery to generate synthetic and natural environments with water body areas. The water bodies in the DRCA2020 dataset are distributed worldwide, as it can be seen in the Fig. 2.

The satellite imagery in the DRCA2020 dataset comes from different public databases. Land surface images are from *ESRI* maps [34], *Bing maps* [35], and *Google maps* [36]; these images are RGB pictures of the Earth's surface. The water coverage maps are from the Global Surface Water database provided by the European Space Agency (ESA) [37]. These maps use blue shades to represent water seasonality. Dark blue represents permanent water, and the lighter blues are areas that only have water during some seasons. The lighter the blue, the less time that area is flooded. The white areas represent land with no surface water. The Digital Elevation Models (DEMs) come from the *ASTER* database, available through the National Space Agency (NASA) *Earthdata* platform [9]. The DEMs are matrices where each value represents the terrain altitude of the point (x, y). The DEM is a one-channel image, which can be normalized to the range of 0 to 255 for visualization purposes. In total, there are eight images for each geographic location (see Fig.3) carefully registered using the QGIS software [38]. For practical purposes, the images collected in the dataset were cropped into patches of 256×256 pixels and converted into PNG format. In total, DRCA2020 contains $13,184$ of these locations, with balanced land and water presence, converting this into a ready-to-use dataset for Machine Learning libraries and algorithms, such as GANs models.

The DRCA2020 collects satellite imagery with terrains on a similar scale and perspective. However, the imagery

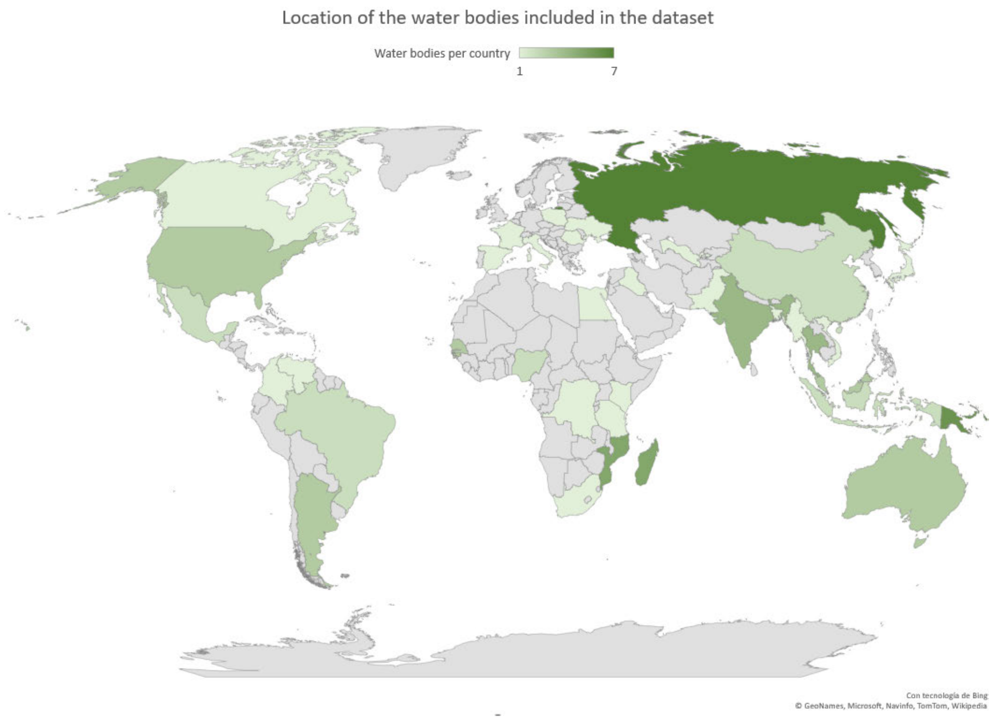

Location of the water bodies included in the dataset
Water bodies per country
1
7
Con tecnología de Bing
© GeoNames, Microsoft, Navinfo, TomTom, Wikipedia

FIGURE 2. Distribution of the river deltas and bays around the world in the DRCA2020 dataset.

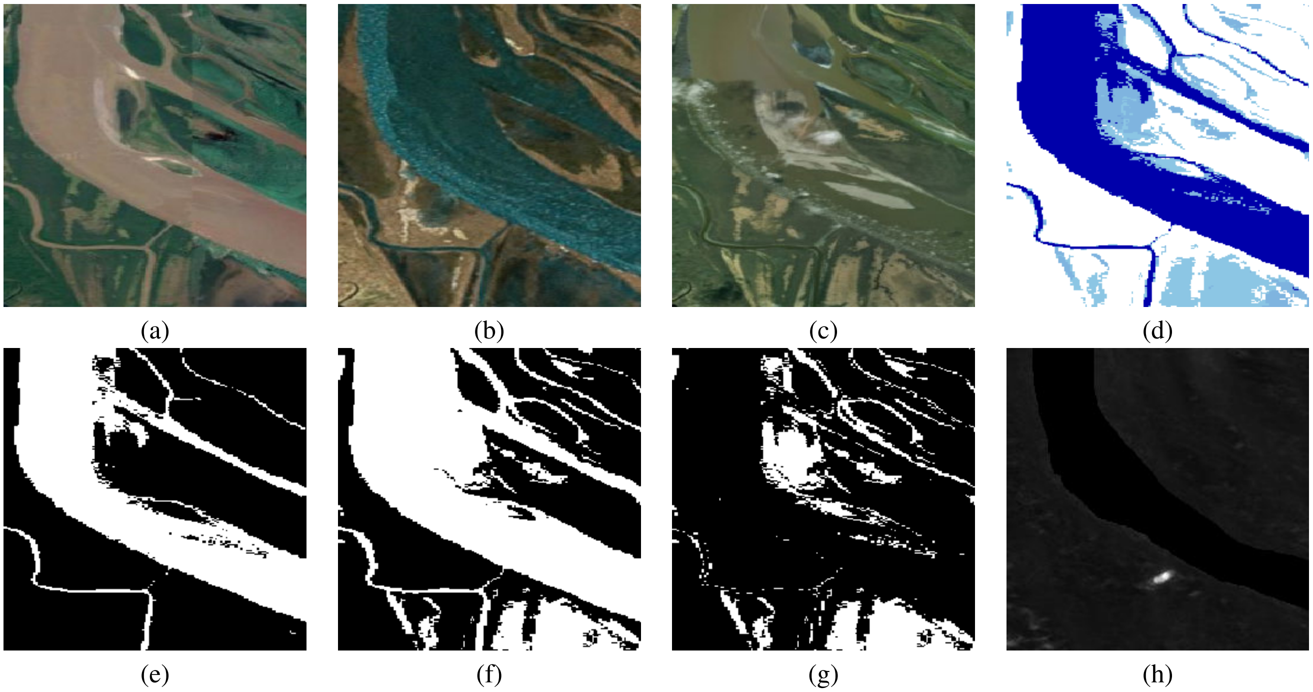


FIGURE 3. Example of image group from the Yukon river: ***Google*** satellite (a), ***Bing*** satellite (b), ***ESRI*** satellite (c), water cover map (d), permanent water (e), total water (f), flooded areas (g), DEM (h).

acquisition is from different time lapses introducing some variability required to avoid overfitting in deep learning model training. One type of overfitting is when the networks start learning noise and interpret it as part of the data [39]; therefore, variations need to be introduced to increase the generalization of the network. In this case, when satellite images are taken at different moments, the variations such as the presence of clouds, the erosion of the coast or even the evolution of the delta channels introduce noise. Therefore, the models need to learn how to generalize the terrain characteristics and avoid repeating the noise.

IV. THE PROPOSED APPROACH FOR TERRAIN GENERATION

The proposed approach consists of three submodules based on the cGANs framework. The first one generates the Digital Elevation Model (DEM) from a water map input. That DEM is now the input for the next submodules which create a land surface image with a polar climate or tropical climate. Note that the user selects either weather or create both land surface images in parallel, see Fig. 1. The cGANs framework is adapted from the works presented in [25] and [26] for this new scenario. Note that the water map is the user's input, which can be in the form of a user-drawn sketch. The surface texture created from a given DEM is an image in the RGB-color model.

Each cGAN is composed of a couple of deep convolutional networks: the generator G and the discriminator D. The generator G learns how to map from an input image x to an output image, this is: y, $G(y) \approx z$. This is different to classic GANs in which G performs the mapping from random noise [40]. In opposition to the generator, the discriminator D tries to distinguish the real image pairs (x, y) from those generated ones $(x, G(x))$. These networks work as adversaries, as G is trained to maximize the classification error that D is trying to minimize. In practice, G will generate images that are closer to z. The networks converge when D is no longer able to tell the difference between real images and the generated ones.

The objective function as established by Isola [26] is:

$$G^* = arg \min_G \max_D L_{cGAN}(G, D)\lambda L_{L1}. \tag{1}$$

The first part in (1) shows the adversarial objectives of the networks as D tries to maximize the differences between real pairs (x, y) and generated pairs $(x, G(x))$, at the same time G tries to minimize that very same difference.

This is more clearly shown in the following equation:

$$L_{cGAN}(G, D) = \mathbb{E}_{x,y}[logD(x, y) + \mathbb{E}_x[log(1 - D(x, G(x)))]]. \tag{2}$$

The second part of the equation (1) is a loss function that relates the generated images $G(x)$ with the expected output y.

$$L_{L1}(G) = \mathbb{E}_{x,y}[\|y - G(x)\|_1]. \tag{3}$$

This loss function $L1$ or least absolute deviation is the sum of the absolute differences between the real image and the predicted image; this is used to improve the generation output as the first part of the equation does not include enough supervision over the generated output. This part of the objective function has a λ multiplier that serves as a weight, in the case of the implementation it was set to 100 to improve the accuracy of the generated images.

On the one hand, the generator's architecture is a U-Net [41] meaning that it has the structure of an encoder-decoder with skip connections. The skip connections add information of the encoding directly to the decoding to decrease loss and improving the results. A U-net uses convolutional layers with 4 × 4 kernels. This kernel size was implemented by Isola. When combined with symmetric padding, even-sized kernels show no shifting problems and yield competitive results in generative networks. The number of filters grow during the encoding part in the following fashion: 64, 128, 256, 512, and then four layers of 512 filters. Each layer uses batch normalization and has a leaky ReLU as the activation function. The decoding involves an up-scaling process either by transposed convolution (or deconvolutional layers) or using up-sampling layers followed by a regular convolutional layer [42]. The proposed approach assessed both up-scaling processes to select the one that reduces artifacts in the generated images. The results of that assessment are discussed in section VI.

On the other hand, the architecture of the discriminator is a PatchGAN [43]. A PatchGan receives patches of the expected image and the generated one to discern and classify the real output of the input image. It uses an Adam optimizer with a mini-batch of one image, and the loss function is the binary cross-entropy. The discriminator and the generator alternate their training, once at a time [7]. This described methodology is an adaptation from the Image-to-Image framework [26].

The three submodules for generation use a similar cGAN architecture using the original setup for every generator and discriminator. That is, the kernel weights were randomly initialized with a normal distribution of median zero and a standard deviation of one. The learning rate is of 0.1 and there is drop out with a probability of 0.1 at each layer in the discriminator network. Additionally, the three cGAN submodules use different training sets for specific purposes. That is, the first one uses pairs of water cover maps as input and a expects a DEM as output. The second and third submodules are for texture generation both with DEMs as input. In this case two different training sets were employed: one uses satellite images of tropical areas whilst the other one uses images of polar areas to generate the outputs. By using these submodules trained with two different areas, it is possible to generate virtual terrains of river deltas and coastal areas and their corresponding textures for both climates.

V. THE FITTING PROCESS

The cGANs that form the backbone of the submodules were trained using the DRCA2020 images. The first cGAN takes water coverage maps as inputs and their corresponding DEMs as outputs. An example of this can be seen in Fig. 4. The following submodules have two different cGANs, one for the tropical climate and the other for the polar climate. These climates are the most predominant in real scenarios. Most of the river deltas worldwide are in tropical climates, while polar climates contain the most significant deltas regarding to their size. In these cGANs the DEMs serve as input while the land surface images are the expected output. An example

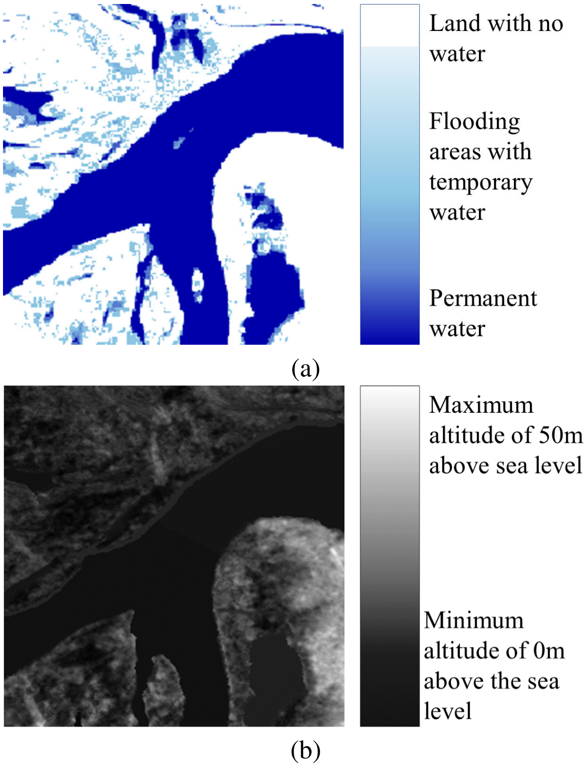

Land with no water
Flooding areas with temporary water
Permanent water
(a)
Maximum altitude of 50m above sea level
Minimum altitude of 0m above the sea level
(b)

FIGURE 4. Images used for fitting the first submodel, with their correspoding color key: a water coverage map (a) is the input, and a DEM (b)is the expected output, which was normalized for visualization purposes(b).

of both pairs of images is in Fig. 5. For visualization purposes, the contrast of the DEMs was enhanced.

A. TRAINING SETUP FOR THE DEM SUBMODULE

The training step for the DEM generator employed 400 randomly selected water coverage maps and their corresponding DEM pairs from the DRCA2020 Dataset. An altitude threshold of 50 meters was established so only DEMs near to the coast could be selected. The training consisted of 150 epochs and was run on Google Colab [44] using a Tesla P100-PCIE-16GB GPU. Note that for visualization purposes the DEMs were normalized into a range of 0 to 255 and the contrast was adjusted in the figures presented in this document.

B. THE TEXTURE GENERATION SUBMODULES

The DRCA2020 Dataset provides the corresponding land surface textures for the water coverage maps and the DEMs pairs. Thence, the training of the texture generators was using the corresponding set of 400 land surface images for both climates polar and tropical. Some different features between the polar and the tropical climates are that the tropical river deltas show bright green forest coverage, while polar landscapes are splattered with small lakes. Therefore, different features require different generative models that fit the corresponding scenario. An additional experiment increased the number of images to 1024 trying to improve the quality of land surface creation. The comparative results are presented in the following section.

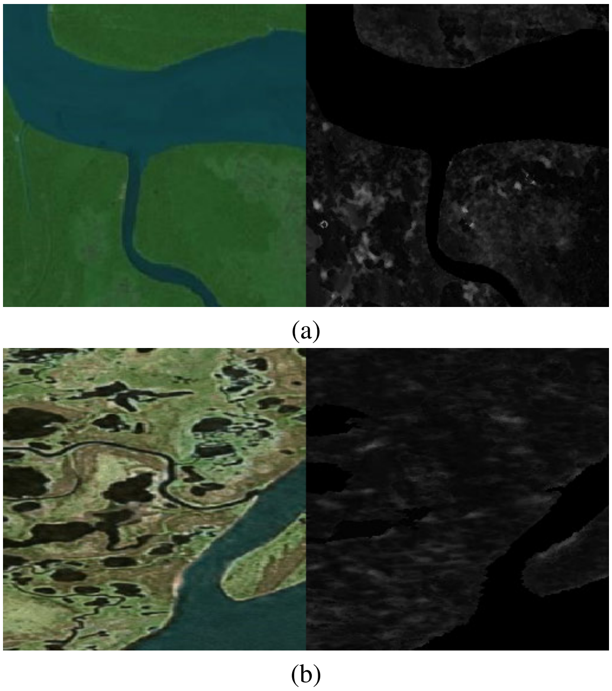


FIGURE 5. Example of a pair of images for the tropical surface generation (a), and for the polar surface generation (b).

The training of the three generators was done under the same Google Colab setup. Additionally, data augmentation was performed to avoid over fitting, specifically random cropping and flipping, as stated in [45].

VI. EXPERIMENTAL RESULTS

This approach generates DEMs and land surface images using a water coverage map as a sole input. The initial training dataset contained 400 images; then, a second training employed 1024 images to assess the visual quality improvement of the generated land surface images. Each experiment took an average of 23 and 71 seconds per epoch respectively when training.

The DRCA2020 dataset collects images of specific geographical locations. The submodule for DEM generation uses the original pairs of water coverage maps and DEMs, th former as input and the later as the ground truth that will be imitated by the generation network. The pairs of DEMs and land surface images are the input and ground truth for the texture generation submodules. By using real-world imagery it is possible to create synthetic realistic DEMs and land texture images. For instance, Fig. 6 shows a pair of DEMs; the image (a), on the left, is a grayscale visualization of a ground truth DEM, while on the right image (b) it is its respectively generated DEM, also visualized as a grayscale image. Fig. 7 shows an example of an user-drawn sketch representing a water map used as input and its generated DEM. Notice that the proposed approach allows manually created drawings that simulate water coverage maps as an input, as well as real-world water coverage maps. The blue color palette represents the flooding areas; the darker, the more time the

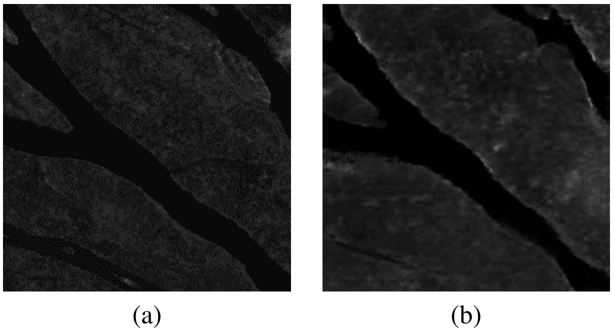


FIGURE 6. A ground truth DEM (a) and a generated DEM (b).

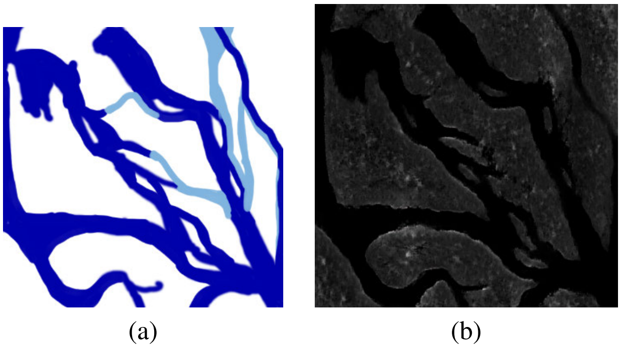


FIGURE 7. User-made sketch of a water map (a), and its generated DEM (b).

area remains flooded, the darkest blue is permanent water bodies. This representation is translated into a difference in altitudes.

The second and third submodules receive a DEM as input and generate a land surface image. Different cGANs were trained to generate either tropical or polar climate surface images, depending on the used training set. Fig. 8 shows the results when using 400 or 1024 training sets for both climates: the input (a) is a real-world water coverage map, which is used to generate a synthetic DEM (b). The generated DEM is the input in the following submodules. Fig. 8 (c) is the generated land surface using the models trained with 400 polar images, while Fig. 8 (d) is when trained with 1024 images. As it was expected, using a larger training dataset with more coastal images results in better preserved details. In counterpart, Fig. 8 (e) and Fig. 8 (f) present the created tropical images with the model trained with 400 and 1024 images. Similarly, using a smaller training dataset generates images with more artifacts.

It is anticipated that the upsampling phase in the U-Net architecture generates some artifacts in solid land areas. In general, two options in the upscaling process are empirically evaluated. The first option is to use a deconvolution layer while the second is to use a upscaling layer followed by a convolutional layer. Due to the specific training dataset in this work, the proposed approach is limited to create images of areas near water. Some artifacts appear in opposite cases, this is, when the images required to be generated are far from the water. In those cases, the deconvolution layer creates a repeating pattern, while the upscaling layer shows vertical and horizontal stripes. Examples of this are shown in Fig. 9 and Fig. 10. The architectures in the three submodules in the proposed approach have the same issue during the upsampling process.

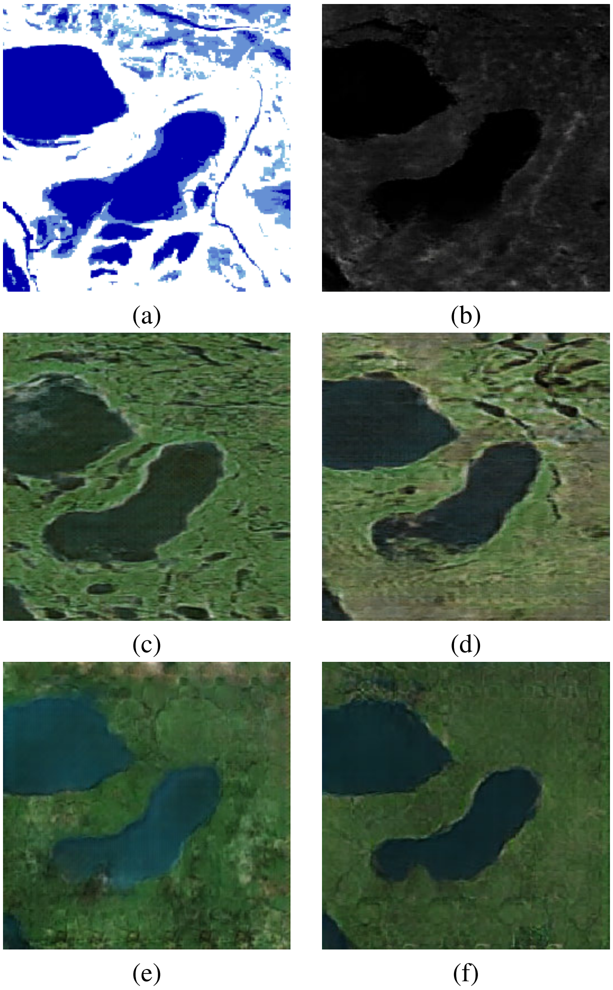
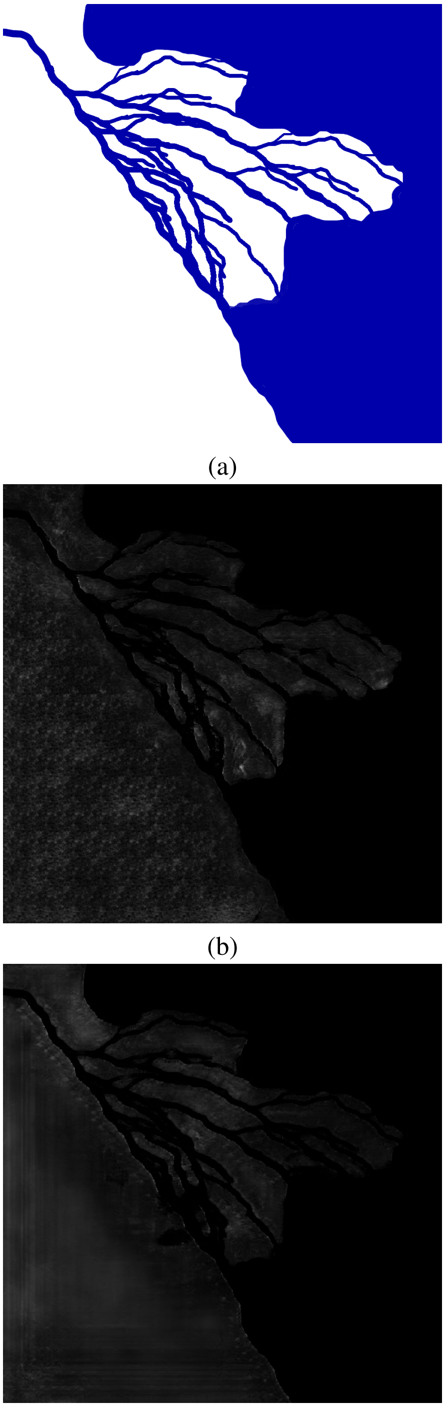


FIGURE 8. Original Water Map (a), the generated DEM (b), generated polar image using a training set of 400 (c) and 1024 images (d), generated tropical images using a training set of 400 (e) and 1024 images (f).

The experimental results show that when deconvolution layers are used in all submodules the checkerboard noise is increased, this can be observed in Fig. 10 (a). In the same way, when the architectures employ upscaling layers, the stripped noise increases, as can be seen in Fig. 10. However, if the DEM is generated using deconvolution layers and that DEM is used in a surface cGAN with upsampling layers, both noises are diminished, this can be seen in Fig. 10 (b). Conversely, if the order of the architectures is swapped, as in Fig. 10 (c), both noises are shown at the same time. Hence, a combination of different upsampling techniques should improve the visual quality results. Nevertheless, a general evaluation of generative models still relies on human perception [46], [47].

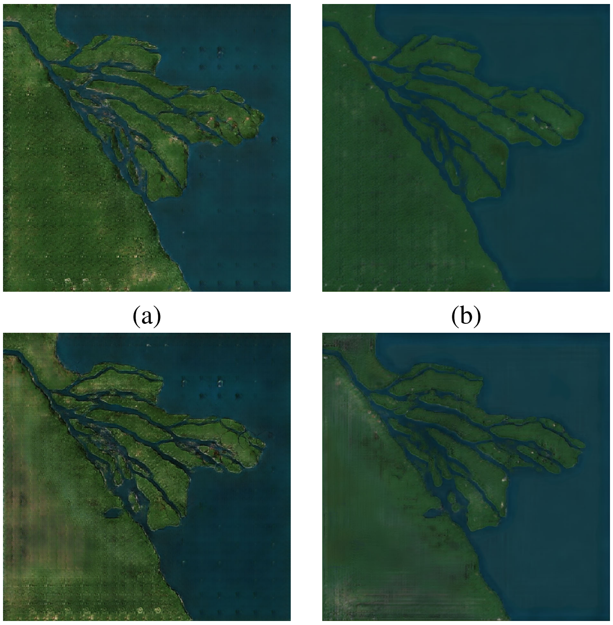


(c)

FIGURE 9. Original Water Map (a), generated DEM using deconvolution layers (b), generated DEM using upscaling and convolution layers (c).

A. COMPARISON RESULTS

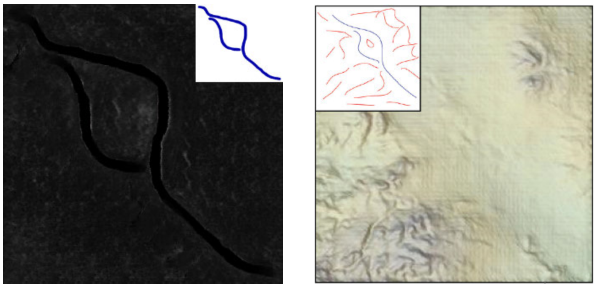
Guérin *et al.* and Teoh *et al.* proposed two different but relevant methods. The former generates height maps from sketch lines representing mountain ridges, river courses, and altitude cues [25]; while the later converts a river mouth into a delta [5]. Until now, there were no methods that create both the height map and the land surface image from a user-made sketch.



(c) (d)

FIGURE 10. Land Surface Images with different combinations of architectures for the DEM and the land surface generation. Deconvolution and deconvolution (a), deconvolution and upsampling (b), upsampling and deconvolution (c) upsampling and upsampling (d).

On the one hand, the similarity between the method proposed in this document and the approach of Guérin is that both use cGANs to generate height maps; however, the one proposed in this document specializes in terrains with water coverage. An example of the comparison of generated elevation models is shown in Fig. 11. Due to our approach's specialization, it can be seen a DEM with a clearer river path created by the proposed method compared with the height map of the related work. In that example, a similar sketch was employed as the input. This qualitative result is generalized with similar experiments.



(a) (b)

FIGURE 11. On the left our approach generates an elevation model with just a single type of feature, although there is some noise in land areas. On the right the approach by Guerin which also needs other features aside from river courses.

On the other hand, the Fig. 12 presents a visual comparison between a river delta generated by the proposed method and the method proposed by Teoh *et al.* Their method focuses

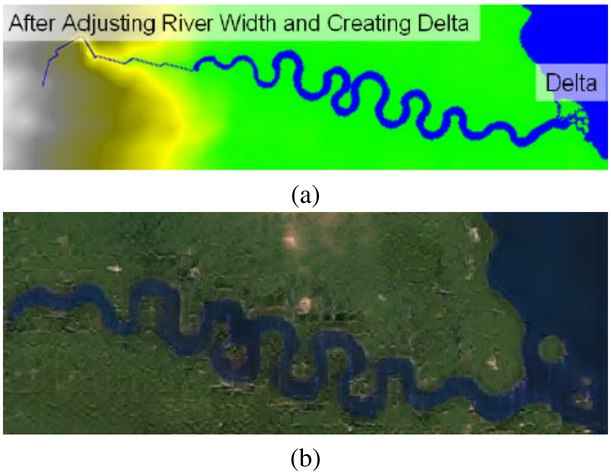


FIGURE 12. A delta generated by the method of Teoh in [5] (a), a delta generated by our approach (b).

on the river geometry to convert a river mouth into a delta. In contrast, our approach can generate an image from an initial sketch. To this comparison, a water map with a similar structure to the river delta generated by the Teoh's method was the input for our approach. In summary, our approach generates terrains with a clearer river path in comparison with other similar method that generates DEMs. Additionally, our approach can also use user-generated sketches as input, which means a high level of control over the generated terrains. Moreover, this approach also generates land surface images that could serve as textures for the generated DEMs if a 3D model of the terrain is to be created from the DEM.

Still, this model's limitation is in generating significant land areas that do not have water bodies or flooding areas. Nevertheless, these cases are not very common in river delta terrains, and the user can address this by adding flooding areas that are translated into lowlands. Mature terrain generation methods work with different terrain features, which usually are mountainous areas. To the knowledge of the authors there are no mature methods that generate river deltas to perform a direct comparison.

Finally, a comparison with real deltas is presented in Fig. 13 with the Fly river delta and Fig. 14 with the Yenisey river delta, neither of these images were used for the fitting process of the cGANs, as these are the full river deltas, as opposed to the training sets which use patches of 256 × 256 pixels. The Fly river is in Papua New Guinea which has a tropical climate, while the Yenisey delta is located in northern Russia and has a polar climate. The original Fly delta is shown in Fig. 13 (a), and its corresponding water coverage map in Fig. 13 (b). The generated delta of tropical climate is shown in Fig. 13 (c) and Fig. 13 (d) shows the delta when generated using the polar trained cGAN. In both cases the structure of the delta channels is respected and the water and land is completely differentiated. As the generation is based on the water coverage map, the clouds that appear on the original image are not present in the generation save in very small spots. In Fig. 14 it can be observed that the delta main channels and islands of the delta are preserved by the submodules with the exception of the smallest ones.

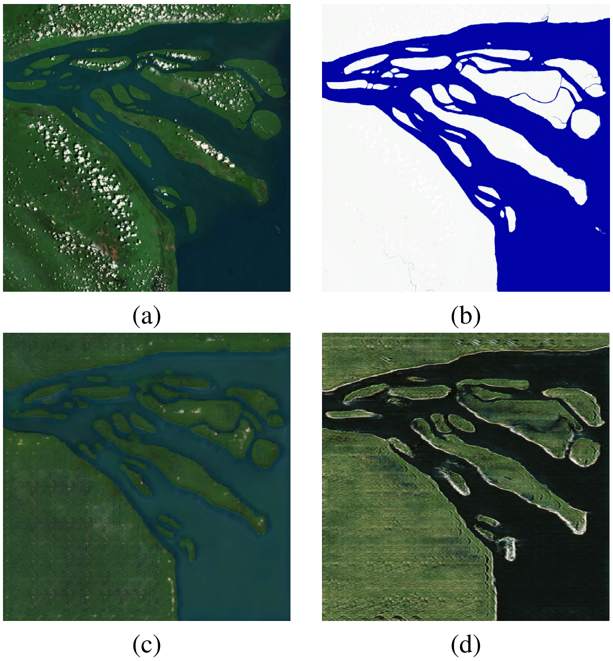


FIGURE 13. Fly river delta (a), its corresponding water coverage map (b), the tropical submodule generation (c), and the polar submodule generation (d).

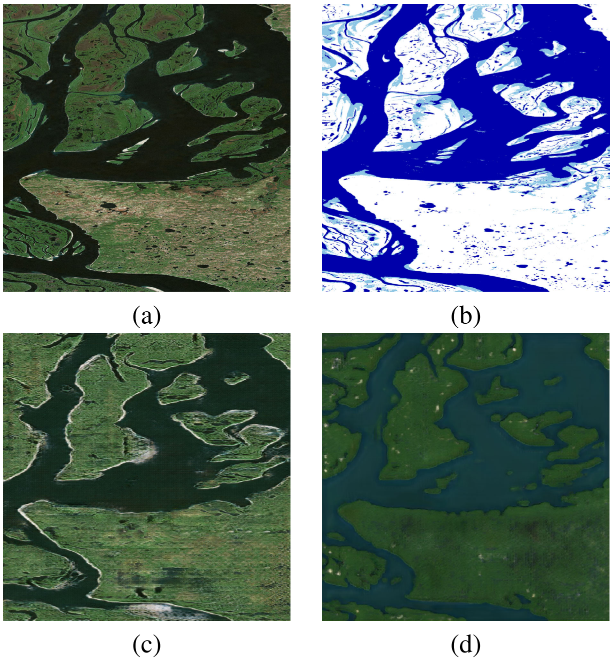


FIGURE 14. Yenisey river delta (a), its corresponding water coverage map (b), the tropical submodule generation (c), and the polar submodule generation (d).

VII. CONCLUSION

This paper presented a modular approach for generating realistic land surface images of river deltas and coastal areas from a water coverage map. The water map or a user-drawn sketch representing a water map is the single required input. The proposed approach ensembles three cGANs submodules to create first, a Digital Elevation Model (DEM) from a water coverage map and then, a group of two cGAN models to create a realistic land surface image from the generated DEM. The proposed approach's training allows creating river deltas based on a polar or tropical climate, or both, with good visual results. This paper also introduced the DRCA2020 dataset with satellite imagery that collects pairs of water maps of important rivers worldwide distributed with their corresponding DEMs, as well as their correlated land surfaces in different climates and time-lapses. The proposed approach contributes to the research area of the automatic generation of water bodies landscapes, providing an ensemble module fitted to this scenario and the specialized dataset to improve the generation process. Particularly, the different land surface textures in the DRCA2020 dataset and hence those created by the proposed approach, introduce variations to the generation of terrains from every single DEM. Moreover, selecting different image subsets from the DRCA2020 dataset to train a deep learning model controls the land surface generation for highly specific scenarios with river deltas in other climates. In counterpart, a constrain of the proposed approach is that introduces some artifacts in flat areas with no presence of water. Those artifacts are due to a lack of training data that include those specific characteristics. Therefore, the modules require to learn flat areas features, which will be addressed in future work.

REFERENCES

[1] R. J. Vitacion and L. Liu, "Procedural generation of 3D planetary-scale terrains," in *Proc. IEEE Int. Conf. Space Mission Challenges Inf. Technol.*, Jul. 2019, pp. 70–77.

[2] G. Lai, W. Latham, and F. Fol Leymarie, "Towards friendly mixed initiative procedural content generation: Three pillars of industry," 2020, *arXiv:2005.09324*. [Online]. Available: http://arxiv.org/abs/2005.09324

[3] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup, "Procedural content generation for games: A survey," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 9, no. 1, pp. 1–22, Feb. 2013, doi: 10.1145/2422956.2422957.

[4] R. J. Hugget, *Fundamentals of Geomorphology*, 2nd ed. Evanston, IL, USA: Routledge, 2007.

[5] S. T. Teoh, "River and coastal action in automatic terrain generation," in *Proc. CGVR*, 2008, pp. 3–9.

[6] E. Nesvold and T. Mukerji, "Geomodeling using generative adversarial networks and a database of satellite imagery of modern river deltas," in *Proc. Eur. Assoc. Geoscientists Eng.*, 2019, pp. 1–5. [Online]. Available: https://www.earthdoc.org/content/papers/10.3997/2214-4609.201902196

[7] J. Langr and V. Bok, *GANs in Action*. Shelter Island, NY, USA: Manning, 2019.

[8] USGS. *Earth Explorer—Home*. Accessed: Feb. 13, 2020.[Online]. Available: https://earthexplorer.usgs.gov/

[9] NASA. *Earthdata Search*. Accessed: Jun. 28, 2020. [Online]. Available: https://search.earthdata.nasa.gov/search/

[10] European Space Agency. *Copernicus Open Access Hub*. Accessed: Feb. 13, 2020. [Online]. Available: https://scihub.copernicus.eu/dhus/#/home

[11] T. N. Mundhenk, G. Konjevod, W. A. Sakla, and K. Boakye, "A large contextual dataset for classification, detection and counting of cars with deep learning," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 785–800.

[12] Z. Liu, L. Yuan, L. Weng, and Y. Yang, "A high resolution optical satellite image dataset for ship recognition and some new baselines," in *Proc. 6th Int. Conf. Pattern Recognit. Appl. Methods*, 2017, pp. 324–331.

[13] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3974–3983.

[14] Y. Liu, J. Yao, X. Lu, M. Xia, X. Wang, and Y. Liu, "RoadNet: Learning to comprehensively analyze road networks in complex urban scenes from high-resolution remotely sensed images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2043–2056, Apr. 2019.

[15] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? The Inria aerial image labeling benchmark," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 3226–3229.

[16] R. Gupta, B. Goodman, N. Patel, R. Hosfelt, S. Sajeev, E. Heim, J. Doshi, K. Lucas, H. Choset, and M. Gaston, "Creating xBD: A dataset for assessing building damage from satellite imagery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2019, pp. 10–17.

[17] P. Helber, B. Bischke, A. Dengel, and D. Borth, "EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 7, pp. 2217–2226, Jul. 2019.

[18] G. Sumbul, M. Charfuelan, B. Demir, and V. Markl, "Bigearthnet: A large-scale benchmark archive for remote sensing image understanding," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2019, pp. 5901–5904.

[19] H. Seybold, J. S. Andrade, and H. J. Herrmann, "Modeling river delta formation," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 43, pp. 16804–16809, Oct. 2007.

[20] D. Justić, N. N. Rabalais, and R. E. Turner, "Modeling the impacts of decadal changes in riverine nutrient fluxes on coastal eutrophication near the Mississippi River Delta," *Ecol. Model.*, vol. 152, no. 1, pp. 33–46, Jun. 2002.

[21] J.-D. Génevaux, R. Galin, E. Guérin, A. Peytavie, and B. Benes, "Terrain generation using procedural models based on hydrology," *ACM Trans. Graph.*, vol. 32, no. 4, p. 143, Jul. 2013.

[22] E. Guérin, J. Digne, E. Galin, and A. Peytavie, "Sparse representation of terrains for procedural modeling," *Comput. Graph. Forum*, vol. 35, no. 2, pp. 177–187, May 2016, doi: 10.1111/cgf.12821.

[23] A. Emilien, P. Poulin, M.-P. Cani, and U. Vimont, "Interactive procedural modelling of coherent waterfall scenes," *Comput. Graph. Forum*, vol. 34, no. 6, pp. 22–35, Sep. 2015.

[24] K. Ping and L. Dingli, "Conditional convolutional generative adversarial networks based interactive procedural game map generation," in *Proc. Future Inf. Commun. Conf.* Cham, Switzerland: Springer, 2020, pp. 400–419.

[25] E. Guérin, J. Digne, E. Galin, A. Peytavie, C. Wolf, B. Benes, and B. Martinez, "Interactive example-based terrain authoring with conditional generative adversarial networks," *Acm Trans. Graph.*, vol. 36, no. 6, p. 228, 0217.

[26] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.

[27] H. Zhou, J. Sun, G. Turk, and J. M. Rehg, "Terrain synthesis from digital elevation models," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 4, pp. 834–848, Jul. 2007.

[28] A. P. Mangra, A. Sabou, and D. Gorgan, "TSCH algorithm-terrain synthesis from crude heightmaps," *Romanian J. Hum.-Comput. Interact.*, vol. 9, no. 2, p. 119, 2016.

[29] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarase, "Learning social etiquette: Human trajectory prediction in crowded scenes," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 549–565.

[30] S. Mohajerani, T. A. Krammer, and P. Saeedi, "A cloud detection algorithm for remote sensing images using fully convolutional neural networks," in *Proc. IEEE 20th Int. Workshop Multimedia Signal Process. (MMSP)*, Aug. 2018, pp. 1–5.

[31] S. Mohajerani and P. Saeedi, "Cloud-net: An end-to-end cloud detection algorithm for Landsat 8 imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2019, pp. 1029–1032.

[32] J. S. Mendes-de Jesus and H. Tomislav. *SoilGrids ISRIC*. Accessed: Feb. 13, 2020. [Online]. Available: http://soilgrids.org

[33] J. M. Coleman, O. K. Huh, and D. Braud, "Wetland loss in world deltas," *J. Coastal Res.*, vol. 1, pp. 1–14, Jan. 2008, doi: 10.2112/05-0607.1.

[34] *World Imagery*. Accessed: May 15, 2020. [Online]. Available: http://www.arcgis.com/home/webmap/viewer.html

[35] *Bing World Map*. Accessed: May 10, 2020. [Online]. Available: https://www.bing.com/maps

[36] *Google World Map*. Accessed: May, 7, 2020. [Online]. Available: www.google.com.mx/maps/

[37] J.-F. Pekel, A. Cottam, N. Gorelick, and A. S. Belward, "High-resolution mapping of global surface water and its long-term changes," *Nature*, vol. 540, no. 7633, pp. 418–422, Dec. 2016.

[38] QGIS Development Team. *QGIS Geographic Information System*. Accessed: Sep. 8, 2019. [Online]. Available: http://qgis.osgeo.org

[39] S. Lawrence and C. L. Giles, "Overfitting and neural networks: Conjugate gradient and backpropagation," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 1, Dec. 2000, pp. 114–119.

[40] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[41] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention* (Lecture Notes in Computer Science), N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham, Switzerland: Springer, 2015, pp. 234–241.

[42] A. Odena, V. Dumoulin, and C. Olah. (2016). *Deconvolution and Checkerboard Artifacts*. [Online]. Available: http://distill.pub/2016/deconv-checkerboard

[43] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *Proc. ECCV* (Lecture Notes in Computer Science), B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 702–716.

[44] *Google Colaboratory*. Accessed: Oct. 28, 2020. [Online]. Available: https://colab.research.google.com

[45] R. Takahashi, T. Matsubara, and K. Uehara, "Data augmentation using random image cropping and patching for deep CNNs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 9, pp. 2917–2931, Sep. 2020.

[46] A. K. Lampinen, D. So, D. Eck, and F. Bertsch. (2018). *Improving Image Generative Models With Human Interaction*. [Online]. Available: https://openreview.net/forum?id=HJr4QJ26W

[47] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4299–4307.

LUIS OSWALDO VALENCIA-ROSADO was born in Campeche, Mexico, in 1989. He received the B.S. degree in international business management and the M.Sc. degree in industrial engineering from the Universidad de las Américas Puebla, in in 2013 and 2015, respectively, where he is currently pursuing the Ph.D. degree in intelligent systems. His current research fields include procedural generation, generative models, and their applications in earth sciences and video games.

ZOBEIDA J. GUZMAN-ZAVALETA (Member, IEEE) received the bachelor's degree in computer science from the Benemérita Universidad Autónoma de Puebla (BUAP), in 2004, and the M.Sc. and Ph.D. degrees in computer science from the Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), in 2008 and 2017, respectively. She is a Mexican Computer Scientist interested in finding relevant and abstract information from complex data, such as video, images, and audio. She has worked as a Visiting Researcher with INAOE, in 2017, and a Visiting Professor with IBERO—Puebla, from 2016 to 2018, and with CINVESTAV-Tamaulipas, Mexico, in 2018. She has been an Associate Professor with the Universidad de las Américas Puebla (UDLAP), since 2019. She is a member of the National Council of Researchers in Mexico (SNI) and the Asociación Mexicana de Computación (AMEXCOMP).

OLEG STAROSTENKO was born in Lviv, Ukraine, in 1959. He received the B.S. and M.S. degrees in computer science from the Lviv State University, Ukraine, in 1982, and the Ph.D. degree in mathematics and physics from the National Autonomous University of Mexico, Mexico, in 1996. Since 1996, he has been a full-time Professor with the Department of Computing, Electronics and Mechatronics, Universidad de las Américas Puebla, Mexico. He has authored more than 200 research articles in several refereed journals, books, and conference proceedings. His current research fields include the access, retrieval, transmitting, and processing of multimedia information in distributed environments. Since 2000, he belongs to the Mexican National System of Researchers (Level I).