

Received October 12, 2021, accepted December 8, 2021, date of publication January 5, 2022, date of current version January 18, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3140226

A Modular Generative Approach for Realistic River Deltas: When L-Systems and cGANs Meet

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

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

Corresponding author: Zobeida J. Guzman-Zavaleta (zobeida.guzman@udlap.mx)

This work was supported in part by the Universidad de las Américas Puebla (UDLAP) Scholarship, and in part by the Consejo Nacional de Ciencia y Tecnología (CONACYT) Scholarship.

ABSTRACT Procedural terrain generation aims to create topographically coherent landscapes with realistic terrain features. Realistic landscapes of our blue planet are not complete without river deltas; however, there is an insufficient advancement in the generation of landscapes with this terrain feature. Therefore, this paper presents a modular approach to generate landscapes focused on the river deltas features. The modular proposal initially creates skeletons of deltas using a stochastic L-system grammar; we include the guidelines for the rules design. In the first module, we propose three L-systems that automatically create delta skeletons using these guidelines. The second module constructs the coastline and the sedimentary lands for the delta skeleton. Finally, the third module uses conditional generative adversarial networks (cGANs) to create the corresponding digital elevation models (DEMs) and land surface images. The evaluation of our proposal includes visual comparisons, and image quality metrics: the Fréchet Inception Distance (FID), and the Naturalness Image Quality Evaluator (NIQE). The proposed modular integration generates realistic deltas with enough variability to outperform related work.

INDEX TERMS L-system grammar, generative model, procedural terrain generation.

I. INTRODUCTION

Procedural Content Generation (PCG) is an area of study that aims to automate the creation of computing assets requiring limited human input. In other words, computer algorithms do most of the content creation process [1]. Many computer assets can be generated automatically, such as characters, music, scenarios, items, stories, landscapes, and virtual worlds, looking for sufficient variability. Terrains are essential assets of a virtual world which are computer representations of landscapes and landforms. In general, procedural terrain generation (PTG) aims to create landscapes with topographic features that are faithful to natural terrains and their variations over time.

PTG research has applications mainly in video game production [2] and virtual reality [3]. Additionally, the application of PTG in data augmentation for Machine Learning has become more prominent because Deep Learning models

require large amounts of labeled inputs during the network training step [4]. Furthermore, Earth system sciences rely on observational data from satellite imagery [5]; the availability of this data is constrained by the satellites' position when capturing images. Thus, generating realistic terrains adapted to those situations can alleviate some limits of data readiness for research purposes.

PTG research focuses on the generation of landscapes with mountains and rivers; however, more complex features, such as river deltas, did not have the same development pace. The complexity of deltas' features comes from their natural formation process when rivers reach bigger and calmer water bodies. Rivers erode the lands they run through and carry the sediment downstream. When the terrain's slope decreases, the current speed also reduces; then, the transported sediment goes to the bottom of the river. Over time, the deposition creates new land [6]. Naturally, these processes depend on the soil composition, rainfall, the slope of the terrain, and the tides and waves on the coast. In particular, if the deposition occurs in the middle of the river, the land cuts the river

The associate editor coordinating the review of this manuscript and approving it for publication was Ramakrishnan Srinivasan^{id}.

channel and forms divergent channels called distributaries. This process repeats itself over time, generating intricate branching shapes. Therefore, river deltas can be classified depending on how this deposition process occurs.

There are different types of branched deltas. On the one hand, river-dominated deltas extend over the receiving water body due to the sediment carried by the river on coasts with no strong waves or great tidal ranges. On the other hand, the tide-dominated deltas grow along the coastal plains with channels widening as they approach the ocean, as a result of ocean water entering during high tides. On the contrary, wave-dominated deltas create a peak-shaped river mouth with no branches because the waves push the sediment back to the existing shoreline; thus, the land created in this way is stratified. Moreover, deltas may present features of more than one type [7]. Figure 1 shows examples of different types of real-world deltas.

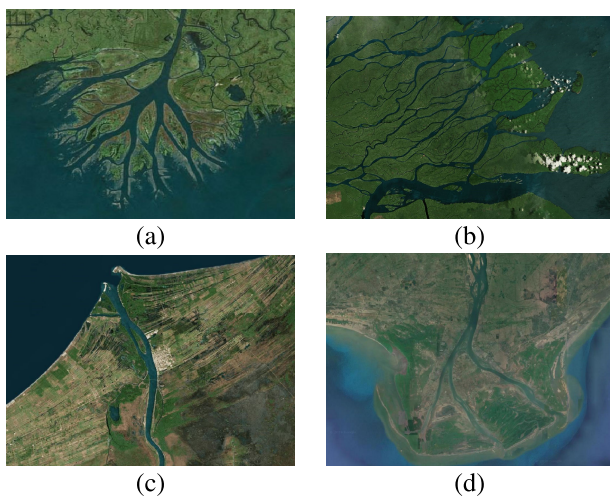


FIGURE 1. Real-world deltas: (a) the Mississippi delta is a river-dominated delta; (b) the Orinoco delta belongs to the tide-dominated type; (c) the Grijalva river delta is an example of wave-dominated deltas; (d) the Krishna river delta presents characteristics from river-dominated deltas and wave-dominated deltas, which are the land expansion and stratification in the coastline, respectively. These images come from Bing maps [8].

In sum, deltas have specific features that differ from the rest of the river. Because of this, current generative models for rivers do not reflect the delta complexity. For instance, some models present ways to generate simplified branching shapes [9]. In counterpart, other models simulating delta formation are highly computationally demanding [7]. Nevertheless, cheaper models based on grammars can reproduce natural structures with fractals, such as the branching shapes in rivers and tide-dominated deltas. In this work, we propose an adaptation of Lindenmayer systems (L-systems) [10] to create skeletons of the river delta based on improvements of our previous work in [11].

The river delta structure is not enough to create a realistic landscape on its own; that is why we propose the integration of generative techniques such as conditional generative adversarial networks (cGANs) and L-systems.

We incorporate the approach in [12] that uses a triplet of cGAN models to generate Digital Elevation Models (DEM) and land surface images using water maps as input.

A water map is an image depicting permanent water in dark blue, seasonal water in lighter shades of blue, and dry land in white [13]. For instance, Figure 2 presents the water maps of some relevant rivers.

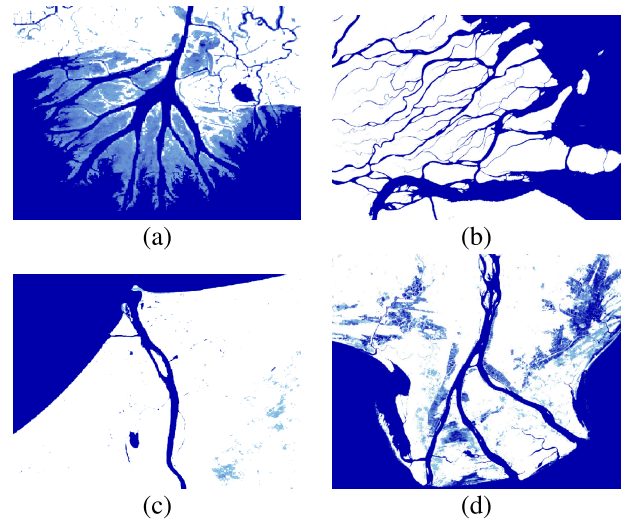


FIGURE 2. Seasonal water maps of some major river deltas. (a) Mississippi, (b) Orinoco, (c) Grijalva, and (d) Krishna. These water maps were obtained from the European Space Agency through its global surface water platform [14].

Thus, this paper proposes a modular method to generate realistic images of the land surface specialized in river delta areas. Our approach integrates an improved stochastic L-system to create delta structures with an image processing module to form their corresponding water map, activating generative models for its land surface image. Moreover, our L-system generates new delta skeletons skipping hand drawing and specializes in the river and tide-dominated types. The ultimate purpose is to automatically create deltas with natural features not existing in the real world, improving the variability of current generative methods.

The remainder of the paper is organized as follows: Section II presents a review of related terrain generation techniques for river deltas. Section III talks about the description of the modules of the proposed method. Subsection III-A presents the module for the generation of river delta skeletons using stochastic L-systems. Subsection III-B describes the image processing module that converts the delta skeletons into water maps. Subsection III-C shows the third module which translates water maps into land surface images. Section IV discusses the experimental results of the entire process. Finally, the conclusions and future work are in Section V.

II. RELATED WORK

PTG research introduced grammar-based methods a long time ago. One of the earliest applications was to rewrite

terrain structures to simulate the erosion processes in small terrains [15]. In particular, a L-system is a parallel rewriting grammar that describes the fractal morphology behavior of biological structures based on initial conditions. Adaptations of L-systems during the interpretation step allow the generation of assets such as trees and bushes. Other computing assets that have been successfully generated with the adaptation of L-systems are roots [16], cities [17], video game creatures [18], and video game levels with their corresponding vegetation [19]. There is an adaptation of L-systems for the generation of delta skeletons in [11]. This method introduces a series of guidelines of the L-system focusing on the sinuosity of the river and its distributaries. Nonetheless, the aforementioned deterministic L-system does not address the lack of variability for more realistic results beyond the initial skeleton.

Other procedural approaches for river delta generation use sequential image processing. For instance, in [7], a simulation-based approach generates realistic delta structures. However, it processes a considerable amount of data, such as water flow and soil features, and requires many iterations to simulate the deposition process. Another method presented in [9], describes a fast and straightforward generation of rivers reaching the ocean. It creates an irregular semicircle of land at the river mouth; then, it generates the shoreline with random points connecting the original mouth to the distributary channels. Thus, this method generates only river-dominated deltas. Other types of methods used in PTG use subroutines with instructions for construction. These instructions have adjustable parameters that allow exerting control over the generated features. The variability of such methods lies in the number of parameters controlled and adjusted by the user. For example, the parameters can control characteristics such as the altitude of mountain ranges and changes in altitude caused by nearby mountains [20]. There are also cases where parameters can control the generation of entire areas rather than individual elements; for example, the type of biomes present in a given latitude and the size of their transition zone into another biome [21]. It is also possible to have several building agents, each with parameters the user can adjust to influence the resulting terrains [22]. Other parameter-driven methods have more significant user interaction, whom chooses the location of specific sections of the generated assets, such as waterfalls and river channels [23]. Finally, other procedural methods try to simulate the natural behavior of water; for instance, in [24], the authors propose a model to animate the movement of ocean waves.

Deep learning models can also generate assets. Current approaches use Generative Adversarial Networks (GANs), composed of a pair of convolutional networks: a generator and a discriminator. The generator tries to create an image belonging to a set of images, while the discriminator is used to identify real images from fake ones [25]. Regular GANs use a random initialization [26], whereas conditional GANs (cGANs) have a specific input such as an image. Providing an initial image to transform into another one is

known as image-to-image translation. For instance, translating a sketch to a colored image, a map to a satellite surface image, and from day-light images to night ones [27]. Image-to-image translation can also be used for problems such as facial aging simulation [28].

In the context of PTG, image-to-image translation learns terrain patterns from a given training input. For instance, in [29], fitted GANs create terrains in the form of a heightmap modeled for video game scenarios. In [30], GANs with images of delta structures as input learned the deposition patterns of deltas. In counterpart, Guérin *et al.* [31] presented a proposal for the generation of mountains and rivers using cGANs with hand-drawn sketches as input for different tasks such as simulating erosion, generating elevation models, and completing missing areas. Finally, the architecture presented in [12] uses water maps as input to generate synthetic digital elevation models and land surface imagery of river deltas fitting three different cGANs. The first one uses a water map as input and generates its elevation model. Then, each of the other two cGANs generates a surface image specialized in tropical and polar climatic zones. However, the need for a natural water map or user-drawn sketches limits the automatic generation capabilities of this method, becoming a bottleneck when many deltas are needed.

III. MODULES DESCRIPTION

The method presented in this paper for delta generation consists of three modules. Figure 3 depicts them. The first module generates a river delta skeleton using a stochastic L-system. The second module uses image processing techniques to create a water map from a delta skeleton. Finally, the water map serves as the input of the third module which generates its corresponding digital elevation model (DEM) and two surface images of different climatic conditions. Each of the three modules is described in the following subsections.

A. THE DELTA SKELETON MODULE

The first module creates a delta skeleton using a L-system. This module improves on the deterministic guidelines previously proposed in [11]. The improvement consists of a) using a new set of stochastic guidelines to create alternative succession rules, maximizing the probability of obtaining valid delta skeletons, and b) establishing ranges for the parameters variation, resulting in a more realistic emulation.

In general, a L-system is a grammar G with parallel rewriting defined by $G = (V, \omega, P)$; where V is the alphabet, ω is the initial state called axiom, and P is a finite set of rules. The alphabet is the collection of symbols available; two or more symbols make up a word. An axiom can be a symbol or a word. The rules consist of the predecessor symbol α and the successor χ , a symbol or a word. For each symbol in a string, the successor replaces the predecessor during the rewriting process according to Equation 1, the result is a new string. Whenever there is no rule for a symbol, the identity is assumed and the symbol is repeated in the rewriting process [10]. The rewriting process is repeated

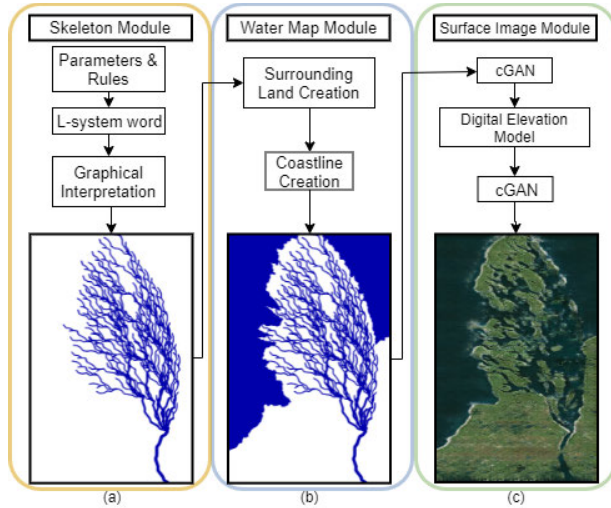


FIGURE 3. The delta generation process is performed by three sequential modules. (a) Module that generates a delta skeleton, (b) module that creates a water map, and (c) module that translates the water map into surface images.

for a number of iterations i .

$$(\alpha \rightarrow \chi) \in P \tag{1}$$

Turtle graphics are useful for providing a graphical interpretation of the resulting strings. Turtle is a vector method that consists of a cursor (the turtle) that draws its route on a Cartesian plane. The cursor parameters are length of the drawn lines r , current position (x, y) , and current direction θ , which can be modified by adding or subtracting an angle β .

The resulting L-system string serves as instructions for the turtle to draw the channels of the delta, therefore, each symbol is a command. The graphic interpretations of the symbols used in this method are explained in Table 1. The push and pop symbols, “[” and “]” respectively, are required to generate branch structures. Symbols that do not have a graphical interpretation are called filler symbols; they are used to initiate the generation, control the generation speed or call a rule under certain conditions.

TABLE 1. The symbols used for the graphical interpretation of the L-system grammar for the generation of delta skeletons.

Symbol	Interpretation
F	Draw a line of size r towards the direction of θ .
+	Direction rotated clockwise by β degrees.
-	Direction rotated counterclockwise by β degrees.
[Current position and direction are pushed to a list.
]	The last position and direction in the list are popped.
I, X, L, R, S	Filler symbols with no graphical interpretation.

1) THE PROPOSED GUIDELINES

It is possible to obtain L-system strings that are interpreted as delta skeletons by following the considerations below. Distributaries are sinuous, this can be emulated when the symbol “F” is surrounded by a couple of “+” and “-”

symbols. What happens is that there is a change of direction, a line is drawn and then the previous direction is restored. An important characteristics of the rules is the balance, it is when within a rule the amount of “+” and “-” symbols is the same; when balance is not kept, the resulting delta spirals as a result.

However, keeping the balance between the symbols of direction change is not straightforward; if two rules that are executed one after the other have the same number of “+” and “-” symbols between the two, there is also a balance, but this tends to result in deltas skewed to one side, however the general direction of the delta is preserved.

The parameter controlling the angle of the direction change (β) should be between 15° and 35° ; smaller angles produce channels with diminished sinuosity that are very close to each other, and larger angles draw angular shapes with no resemblance to deltas. Figure 4 shows some examples of the effects on the resulting skeletons caused by the different modifications on the rules and parameters.

It is also necessary to set the values of the initial parameters. We propose different ranges for a probabilistic selection of them to add variability to the generated shapes. Table 2 summarizes the proposed intervals obtained empirically.

TABLE 2. Initial parameters and values intervals.

Parameter	Value range	Effect
Direction change β	[15, 35]	Delta shape preserved within these values.
Step size r	[11, 20]	Drawing within these margins provides detail and complete deltas.
Starting position (x^0, y^0)	(0, [300, 700])	The delta origin has a variable position around the center of the left border of the image.
Starting direction θ^0	[-20, 80]	Initial direction, given the initial position is on the left border of the image.
Iterations i	[5, 11]	More iterations create more intricate structures, an interval of 3 values should be chosen by the user depending on the L-System.

In accordance with the above constraints, we propose the following guidelines to add variation to the rules while keeping balanced growth of L-systems for the creation of delta skeletons:

- Swap the position of two opposite direction change symbols.
- Add a pair of direction change symbols separated from each other within a rule.
- Add a single line draw symbol F or axiom symbol I .
- Change the position of a filler symbol, as long as this does not leave a pair of push and pop symbols empty.
- Eliminate pairs of direction change symbols, as long as enough of these symbols remain within the system to maintain the sinuosity.

Using these guidelines to create rule variations, we have constructed three stochastic systems as valid examples. Those

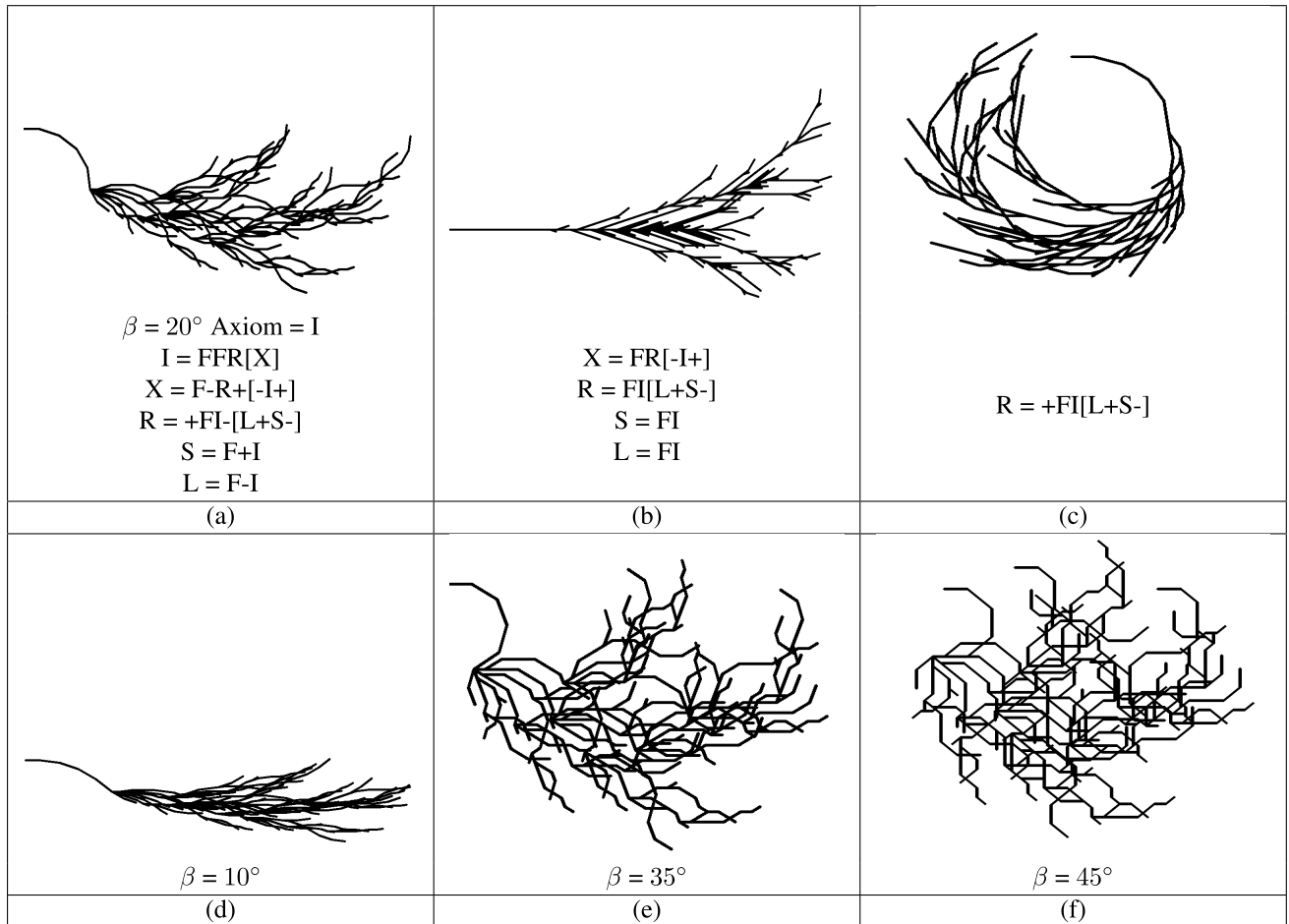


FIGURE 4. Variation effects in a L-system. (a) The base L-system, proposed in [11]. (b) Example of variation; there is no direction change symbols surrounding the line draw symbols, thus, sinuosity is lost. (c) When the + and - symbols are uneven in the rule R, the delta spirals. (d) When β is below 15° , the resulting delta loses many features. (e) The upper limit for β (35°). (f) If β is too big, the delta shape is lost.

TABLE 3. The first stochastic L-system generates deltas with intricate channel structures due to the axiom's presence in all the rules, effectively restarting the system at each iteration. A delta that comes from a word created by the I, X, and R rules with fewer direction change symbols would show less sinuosity in its channels.

Stochastic L-System 1	
Rules	Parameters
$I \rightarrow F + F - -R[X]$ $I \rightarrow FFR[X]$	Axiom = I
$X \rightarrow FR[I]$ $X \rightarrow F - R + [-I+]$	Iterations $\in [6, 9]$ $\beta \in [15, 35]$
$R \rightarrow +FI - [LS]$ $R \rightarrow +FI - [L+S-]$	$r \in [11, 20]$
$S \rightarrow F + I$	$x_0 = 0, y_0 \in [300, 700]$
$L \rightarrow F - I$	$\theta \in [-20, 20]$

instances are presented in the tables 3, 4, and 5. In the first system, in Table 3, the first variation in the rules eliminates the direction change symbols in the rule for the symbol I; all possibilities for the rules X and R include balanced direction

TABLE 4. The second stochastic L-system presents noticeable self-similarity due to the axiom's presence in the successor of its own rule. The number of F symbols determines the growth rate of this system.

Stochastic L-System 2	
Rules	Parameters
$I \rightarrow -F + F[-RI] - F + F[+RI] - F +$ $I \rightarrow -F + F[+RI] - F + F[-RI] - F +$ $I \rightarrow -F + F[-R] - F + F[+R]$	Axiom = I
$R \rightarrow F - LSF - F + F$ $R \rightarrow F - SLF - F + F$ $R \rightarrow F - LSF - F$	Iterations $\in [6, 9]$ $\beta \in [15, 35]$
$S \rightarrow F - I$ $S \rightarrow F - I+$ $S \rightarrow F - I-$	$r \in [11, 20]$ $x_0 = 0, y_0 \in [300, 700]$
$L \rightarrow F + I$ $L \rightarrow F + I-$ $L \rightarrow F + I+$	$\theta \in [10, 50]$

change symbols, while the rules S and L complement each other. This balances the final word produced by the L-system. The system presented in Table 4 has three possible rules for each symbol. In all cases, the positions of the direction

change symbols were modified. There are rules for S and L that are not balanced; this skews the system to one side but does not make it spiral because the system does not present other imbalances. The system grows rapidly due to the size of the rule for I and how often this rule is repeated. Furthermore, the final L-system presented in Table 5 shows swaps in change direction symbols, as well as filler symbols exchanging positions. Thus, this system does not present any imbalance in its rules.

TABLE 5. The third stochastic L-system has fewer branching than previous systems since only the rule R produces it. The direction change symbols surrounding the axiom in the rules S and L will create entangled channels in some of the combinations.

Stochastic L-System 3	
Rules	Parameters
$I \rightarrow +F - F - RX$ $I \rightarrow -F + F - RX$ $I \rightarrow -F - F + RX$	Axiom = I
$X \rightarrow +F - F + RI$ $X \rightarrow -F + F + RI$	Iterations $\in [8, 11]$
$R \rightarrow F[SL]F$ $R \rightarrow F[LS]F$	$\beta \in [15, 35]$ $r \in [11, 20]$
$S \rightarrow F - I+$ $S \rightarrow F + I-$	$x_0 = 0, y_0 = [300, 700]$ $\theta \in [20, 60]$
$L \rightarrow F + I-$ $L \rightarrow F - I+$	

Stochastic L-systems aim to produce different deltas each time they are executed. The number of systems that a stochastic L-system can produce is given by the multiplication principle in Equation 2; where L are the possible structures that will be generated by a stochastic L-system, N is the number of symbols, and x is the number of rules each symbol has. Aside from this, the graphical interpretation is also modified with probabilistic parameters for the initial position, direction of the delta, the direction change, the step size when drawing lines, and the number of iterations. Finally, the deltas can be rotated or mirrored once finished. Each rule has the same probability of being chosen. Thus, by following the guidelines it is possible to create L-systems with more rules and achieve more variability.

$$L = \prod_{i=1}^N x_i \tag{2}$$

B. THE WATER MAP MODULE: FROM SKELETONS TO WATER MAPS

The skeleton is the graphical interpretation of the L-system and resembles a river delta by itself; however, the receiving water body has not been defined yet. Therefore, the water map module uses an algorithm to create natural-looking contours for irregular objects, generating the sedimentary land surrounding the delta channels and the boundary for the coastline. That is, the second module converts the delta skeletons

generated by the L-systems into water maps. A diagram of the process of creating the water maps is in Figure 5.

This process is subdivided into several subroutines:

- 1) Generation of delta’s surrounding land:
 - Creation of convex hull to simulate the surrounding land
 - Computation of control points to generate coarse edges
 - Refinement of edges of surrounding land
- 2) Generation of the coastline:
 - Creation of the main features of the coast
 - Refinement of the coast
- 3) Integration of surrounding land and coastline with the delta skeleton to form the water map

The process for creating the water map generates the surrounding land, the delta channels, and the coastline on a blue background. The land surrounding the channels and the coastline are polygons described by the set of points A and B , respectively. The generation of these two sets is found in the following subsections.

1) SURROUNDING LAND

The first step is to determine the convex hull of the delta skeleton, which is the smallest convex polygon surrounding a set of points. The set A is initially formed by the vertices of the convex hull. The OpenCV function for finding contours computes the convex hull. Figure 6 shows an example of a delta skeleton with its convex hull filled in white surrounding it.

Algorithm 1 Segment Subdivision for the Fitting of Delta Channels

```

Input: Set of points A
Output: Subdivided set of points A
1: for  $p \in A$  do
2:   for  $j \in [1, \lfloor D_p/Z_i \rfloor - 1]$  do
3:      $x_j = x_{p_a} * (1 - j / \lfloor D_p/Z_i \rfloor) + x_{p_b} * j / \lfloor D_p/Z_i \rfloor$ 
4:      $y_j = y_{p_a} * (1 - j / \lfloor D_p/Z_i \rfloor) + y_{p_b} * j / \lfloor D_p/Z_i \rfloor$ 
5:     if  $(i \neq 1)$  then
6:        $x_j + \rho_x$ ; where  $\rho_x$  random  $\in [-Z_i/2 \dots Z_i/2]$ 
7:        $y_j + \rho_y$ ; where  $\rho_y$  random  $\in [-Z_i/2 \dots Z_i/2]$ 
8:     end if
9:     Add the new point  $(x_j, y_j)$  to the set A
10:   end for
11: end for
12: return A
    
```

In order to fit this initial polygon into the delta channels, more points are added to the set A by subdividing each line segment formed by a pair of adjacent points $p = (p_a, p_b)$. The subdivision procedure is shown in Algorithm 1; where D_p is the Euclidean distance between adjacent points; Z_i is the desired sub-segment size at the current subdivision i ; j is the current created point, and $j / \lfloor D_p/Z_i \rfloor$ is used as the weight value for the linear combination between the points p_a and p_b to obtain the Cartesian coordinate of j , i.e. (x_j, y_j) . The last part randomly moves the point around its original

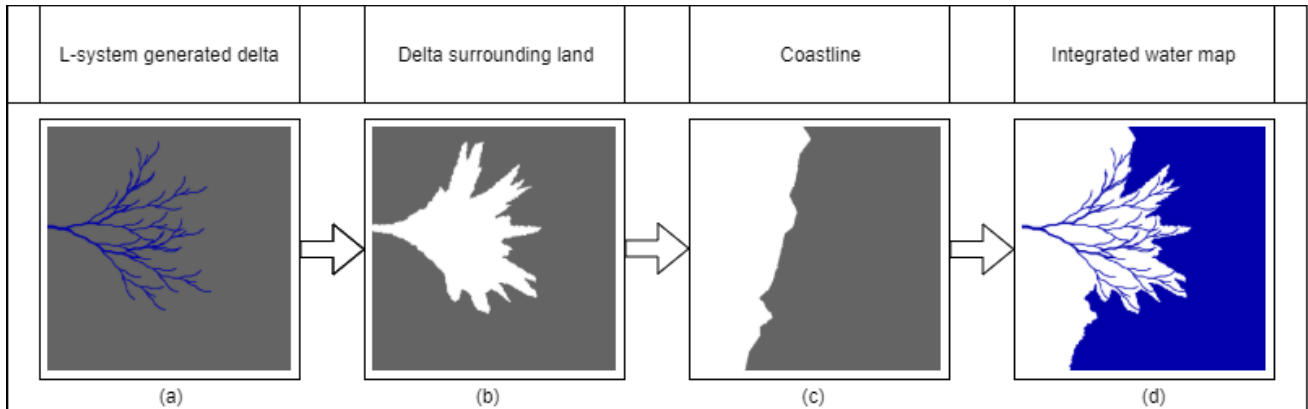


FIGURE 5. Water map creation process. (a) A delta skeleton is the input, (b) generation of surrounding land, (c) generation of the coastline, (d) integration of the surrounding land, the coastline and the delta skeleton into a water map.

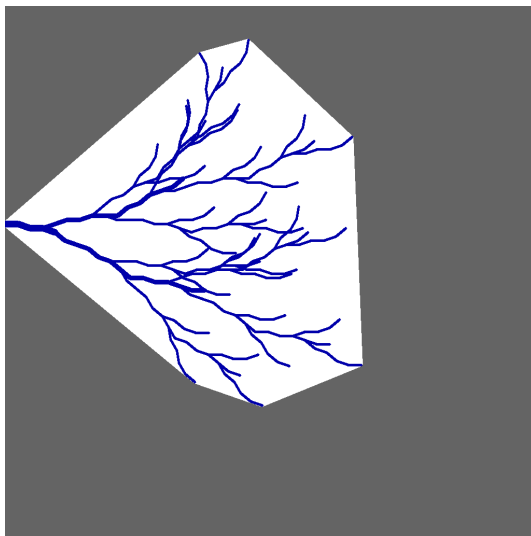


FIGURE 6. Example of the convex hull of a delta skeleton.

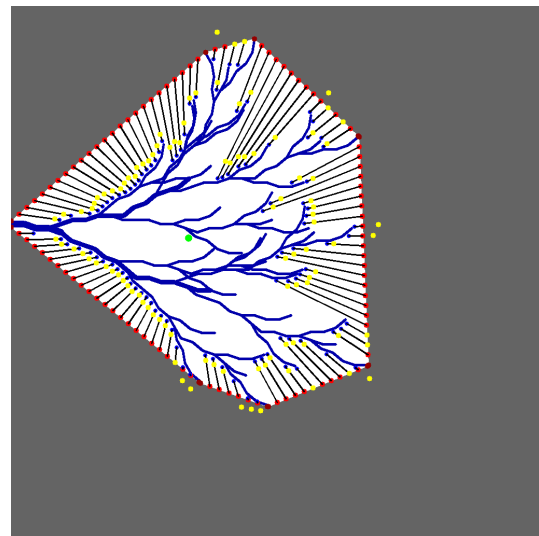


FIGURE 7. Fitting the convex hull. The new points on the hull are shown in red, the black lines show the exploration path towards the centroid, which is in green. The yellow points represent the new position of each point after the fitting process.

position in a neighborhood of $Z_i + 1 \times Z_i + 1$ pixels; the size of the neighborhood was chosen to avoid overlapping points intersecting each other when it is larger, it should be noted that a smaller neighborhood often does not generate enough sinuosity. This last step is not performed in the first subdivision.

For the first subdivision of the set A , the value of Z_1 is 20 because it allows creating enough controllable points to adjust the initial land to the channels. The value of Z_1 was defined through empirical experimentation when considering an image of size 1024×1024 pixels. The value assigned to the other Z_i was also found empirically. The resulting points after the first subdivision are depicted in red in Figure 7.

The following step adjusts the position of the points in A to the delta channels. This process explores the space between each point in A and the centroid of the convex hull; the exploration is carried out in a straight line considering a neighborhood of 7×7 pixels because sometimes the closest

channel is not exactly on the exploration path. The position of each point in A is updated based on the three possible exploration scenarios:

- 1) No channel is found during the exploration. In this case, the point moves towards the centroid, at 90% of the original distance between the point and the centroid.
- 2) The channel is close to the original position of the point in A , this is, at 10% or less of the distance between the point and the centroid. In this case, the point is randomly moved in the opposite direction between 5% to 10% of its distance from the centroid. The land is increased in these cases simulating the deposition that surrounds the channels.
- 3) The channel is not near the point. In this case, the point approaches the position of the delta channel, leaving enough space for the land surrounding the channel.

The new position of the point will be between 90% and 95% of the original distance between the point and the channel. This allows the new shore to be located close to a delta distributary and at the same time prevents the surrounding terrain from disappearing.

The relocated points can be observed in yellow in Figure 7, the black lines are the exploration trajectories of each point without the neighborhood. This set of points describes a coarse shape around the delta channels, which needs to be refined to obtain sinuous lines. To achieve this, the subdivision routine is performed one more time. Z_2 takes the value of 10 pixels because the sinuosity is expected to be subtle but still visible, and this time the step of relocating the point within the neighborhood is performed. The resulting set A describes a polygon that represents a more natural-looking depositional land around the delta channels. An example of the resulting refined shape is shown in Figure 8.

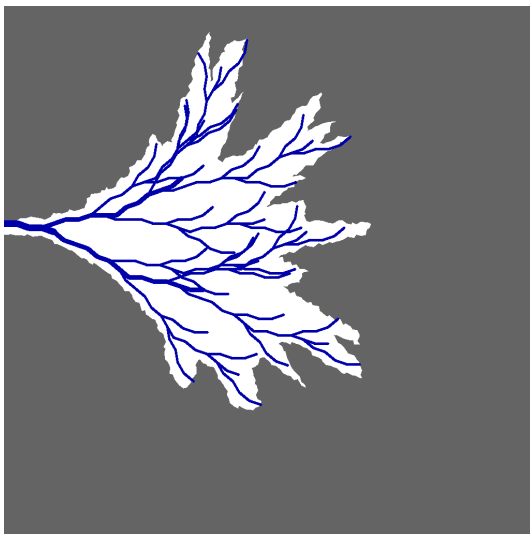


FIGURE 8. The land boundary of the delta is refined to look more natural. This polygon is defined by the final set of points A .

2) CREATION OF THE COASTLINE

The second set of points B describes the polygon that forms the coast. The first point that belongs to B is the coastline origin; finding it involves using the delta origin and the point on the convex hull that is furthest from the delta origin. A linear combination of these two points determines the position of the coastline origin point; the weighting value will determine how close the coastline origin is to the delta origin, which is controlled by the user and is called λ . This is used to create river- or tide-dominated deltas using the same skeletons. When the value of λ is close to 1, the coastline is close to the origin of the delta, the resulting shape resembles a river-dominated delta where the delta grows over the receiving body of water while creating land surrounding its branches. When the value of λ is close to 0, the coastline will be near the furthest point on the convex hull and the coast will cover most of the delta, this resembles a tide-dominated delta. In this

type of deltas the sediments are pushed back by the action of the tides resulting in a more homogeneous accumulation of land along the coastline and the delta extends over the coastal plains. An example of the differences between river- and tide-dominated delta types is shown in Figure 9, sub-figure (a) uses a λ of 0.8 while in sub-figure (b) it is of 0.2.

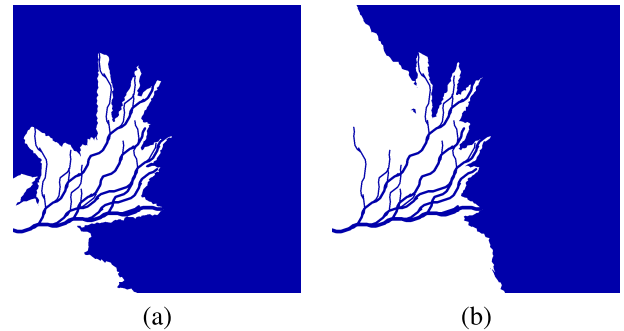


FIGURE 9. Example of delta types. (a) A delta dominated by the river grows over the receiving water body. (b) A delta dominated by the tides where the land formation expands the shoreline more evenly. Both delta types can be simulated by our approach.

After selecting the coastline origin, the upper and lower limits of the coast are searched. Three points are needed: the delta origin, the coastline origin, and a third point, which has the value x of the coastline origin and the value y of the delta origin. These three points form the angle γ , which has its vertex at the delta origin, then its amplitude is calculated. The angle indicating the position of the lower coastal limit, γ_L , is obtained by adding 70° to 110° to γ ; this range is used to add variability. The angle corresponding to the upper coastal limit, γ_U , is obtained by adding 160° to 200° to γ_L . Using these angles, the coordinate positions of the upper and lower coastal limits can be calculated.

The initial set B is made up of these three points: the coastline origin, its upper limit, and its lower limit. An example of this initial coastline is shown in Figure 10. The convex hull is shown filled in white, the delta origin is shown in blue and the furthest point on the convex hull is in green. Within this segment, a linear combination using a λ of 0.8 determines the position of the coastline origin. Then the two points defining the upper and lower limits of the coast are used to draw the line segments that describe the initial coastline.

The two segments formed by the coastline origin and the lower and upper limits are subdivided to create variations and accidents. For this purpose, Z_3 takes a value of 50, because the intention is to create bigger accidents in the coast than in previous subdivisions. These initial accidents look sharp and angular; to improve on this, the coastline is refined by performing another subdivision, this time using Z_4 , which is assigned a value of 10. After this, the resulting set B describes a more refined coastline. The set B is finalized by adding the image vertices that lie within the coast; these are found by moving clockwise from the upper coastal limit until arriving to the position of the lower coastal limit. Now the set B

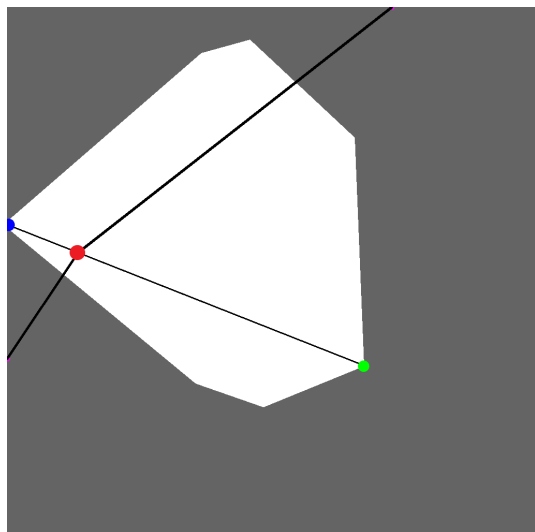


FIGURE 10. The delta origin (blue) and the furthest point on the delta hull (green) are used to find the coastline origin (red). From there, the upper and lower coastal limits are found.

describes a closed polygon. An example of a coarse and a refined coastline is shown in Figure 11.

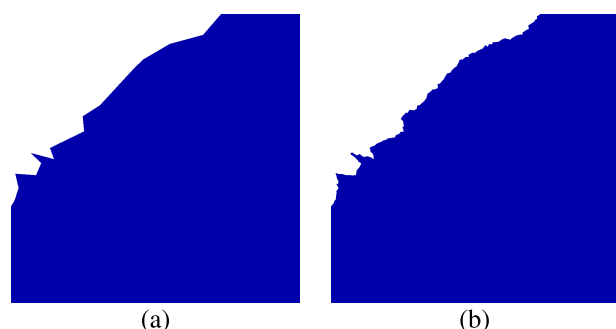


FIGURE 11. Refinement of the coast. (a) Coarse accidents of the initial coast. (b) Coast refinement to be used in the water map.

To finalize the water map generation, an image is filled in blue ((0, 0, 170) in the *RGB* color model), and both sets of points *A* and *B* are drawn and filled in white to represent all the land. Finally, the delta is drawn on top of this using the same shade of blue.

C. THE ELEVATION MODEL AND SURFACE IMAGE MODULE

The third module uses the method proposed in [12] to generate digital elevation models (DEMs) and realistic delta surface images. The cGANs approach proposes to use as input either a user-drawn sketch or a real-world water map. However, this paper introduces an automatic process for generating original river deltas without manual intervention using the water maps from our proposed first two modules. The generated water maps are used as input for the original cGANs, which were trained and validated using the DRCA2020 dataset, publicly available in <https://github.com/DRCA2020/Tropical-Rainforest-and-Monsoon>. The cGANs

are trained with pairs of images that serve as input and expected output. The first cGAN uses a pair of water map and DEM; then the climate cGANs are trained using pairs of DEM and tropical surface images, and DEM and polar surface images respectively. [12]. Figure 12 shows the generation pipeline presented in [12].

The cGANs approach uses a water map as input and generates a digital elevation model (DEM), which is a matrix containing the information of the terrain altitude in meters at each position (x, y) . Figure 13 shows an example of a DEM, normalized into a grayscale image for visualization purposes. Then, from each DEM two cGANs can generate surface images corresponding to deltas in tropical and polar regions respectively. In sum, the cGANs approach generates the elevation model and two distinct surface images from a single water map input.

IV. RESULTS AND DISCUSSION

This section presents some river delta generation examples showing the L-system skeleton, water map, DEM, and surface images. Then, we provide the quality assessment of the generated images. Finally, this section shows a comparison against other related works.

A. THE GENERATIVE PROCESS RESULTS

As it was described in section III, each stochastic L-system is capable of generating multiple skeletons given the same parameters. Two examples of the generated skeletons are presented for each of the three L-systems that were designed. Figures 14 and 15 show an example of the generation results using the L-system 1. In both cases, sub-figure (a) shows the skeleton generated by the L-system. The resulting image in the Figure 14 was rotated 90° counterclockwise. This delta has intricate distributaries due to the effects of rules that call the axiom several times during the rewriting process. The generated delta, as can be observed in sub-figure (b), is a river-dominated one. However, its surface images have characteristics of the tide-dominated type, such as the size of the channels and the apparent incursion of the sea in the delta area. The channels' widths diminish as they move further away from the delta origin. The image in Figure 15 (a) shows intricate braiding channels. In sub-figure (b), the coastline origin covers a big portion of the delta and the width of the channels. It is a tide-dominated delta. The sub-figure (c) shows the result of the DEM generation with the first cGAN. The result of the polar surface cGAN is shown in sub-figure (d) and the tropical one is in sub-figure (e). In the case of Figure 14 the result exhibits sandbanks along the islets created by the delta. Despite both were generated from the same system, the differences are obvious; the first one presents entanglements in its channels and is skewed towards its left side, while the second delta is only lightly skewed upwards and does not present entanglement.

Figures 16 and 17 show the results generated by the second L-system. Both deltas show less and more spread channels than system 1; this happens because branching symbols in

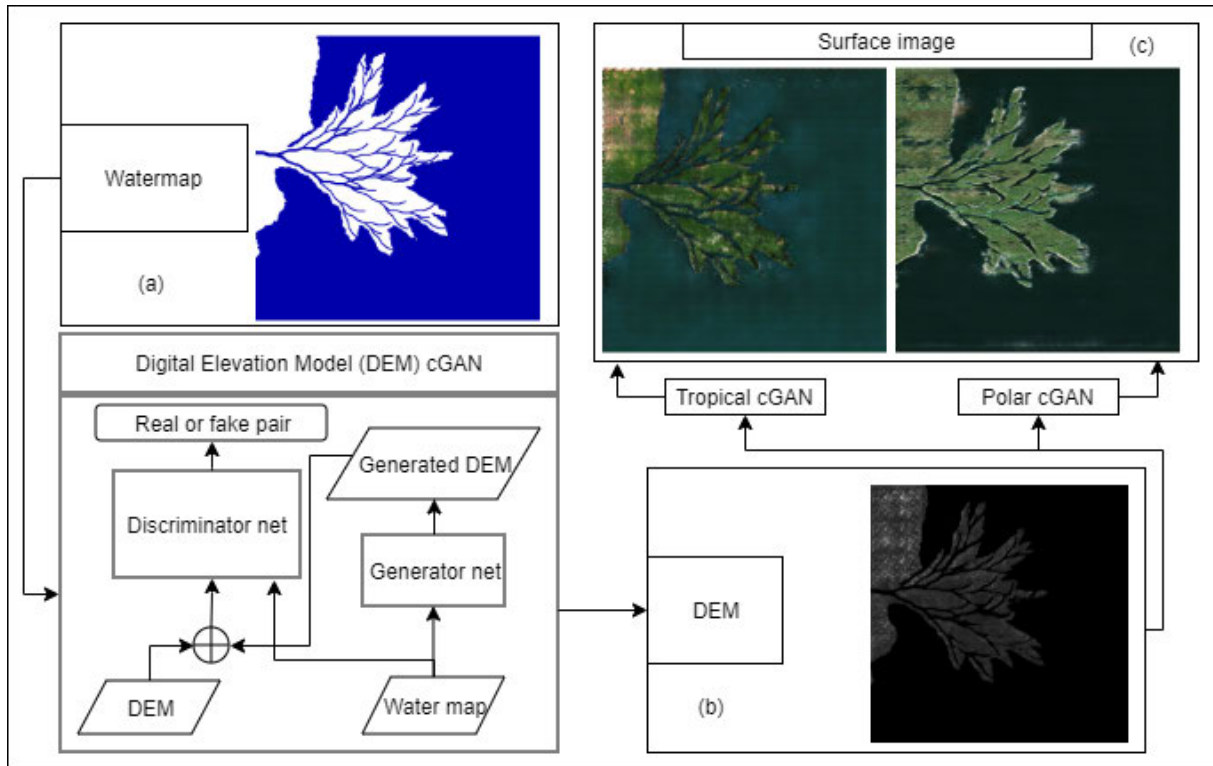


FIGURE 12. The third module, previously proposed in [12], uses three cGANs. The first one takes a water map as input (a) and converts it to a digital elevation model (b). Then, the user chooses between the other two cGANs to generate a tropical landscape or a polar one respectively (c).

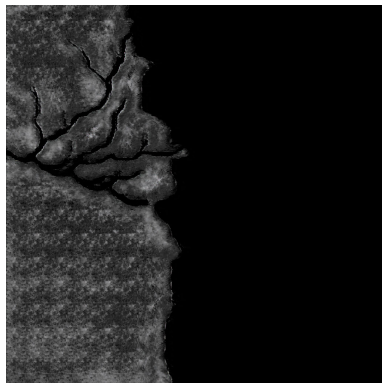


FIGURE 13. A digital elevation model (DEM). For visualization purposes, it can be represented as a grayscale image where the lowest areas are in black and the highest in white. A black value represents an altitude of 0 and a white is a maximum altitude set to 50 meters; assuming a near to sea level altitude.

the rules are more limited than in the previous system. The resulting deltas are river-dominated as well. The delta in Figure 16 has more straight channels in comparison with the one in Figure 17 due to rules with fewer direction changes. The self recurrence is apparent in both deltas, showing that small variations in the L-systems allow preserving general features of the generated deltas. Even more, Figure 17 resembles the Mississippi delta shown in the sub-figure (a) of Figures 1 and 2. The polar and tropical images preserved the channels generated by the L-system.

Following that, Figures 18 and 19 show two different deltas from the third L-system. The first one has a skewed upper portion, while the second one is almost symmetric. The braiding channels in Figure 19 are achieved by surrounding axiom generation symbols with change angle symbols recurrently. This created image resembles the Krishna river delta, which has only three distributaries and a braided section, as it was shown in sub-figure (d) of Figures 1 and 2. In these two deltas, the coastal origin is very close to the furthest part of the delta, and the result is that the coastline is in the ending of the channels. The result is a delta that is dominated by the tides and waves.

The above examples show the variability in the resulting deltas; even if generated by the same system, the results are very different due to the randomness involved in generating the water map and the skeleton. For this reason, the systems and parameters should be saved for delta reproduction. The proposed method presents a significant advantage, as it can generate deltas without needing user input in the form of an initial water map. It can automatically generate original river deltas and their corresponding surface images along with the elevation information. Previous proposals lack enough variability when generating the delta skeletons, as a deterministic system generates just one skeleton. Even more, with the proposed guidelines for designing additional rules for each symbol, it is possible to increase the generative capabilities of the same L-system.

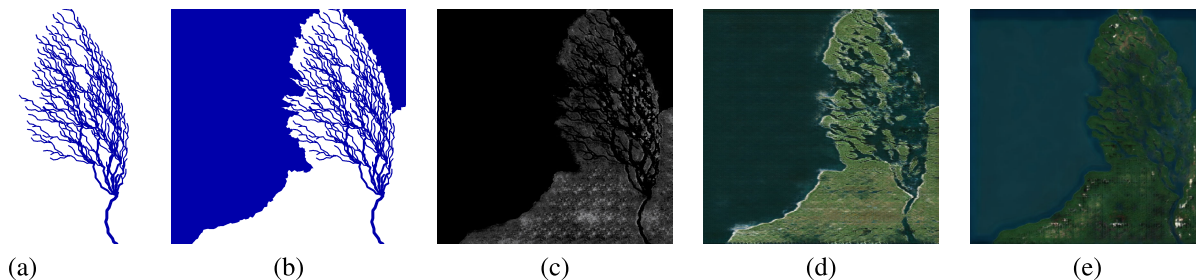


FIGURE 14. A delta structure using the first L-system of Table 3. (a) shows the graphical result rotated 90° clockwise, and (b) is its corresponding water map. (c) The height map generated by the first cGAN; (d) and (e) show the polar and tropical climate images. These were generated by independent cGANs using the height map as input.

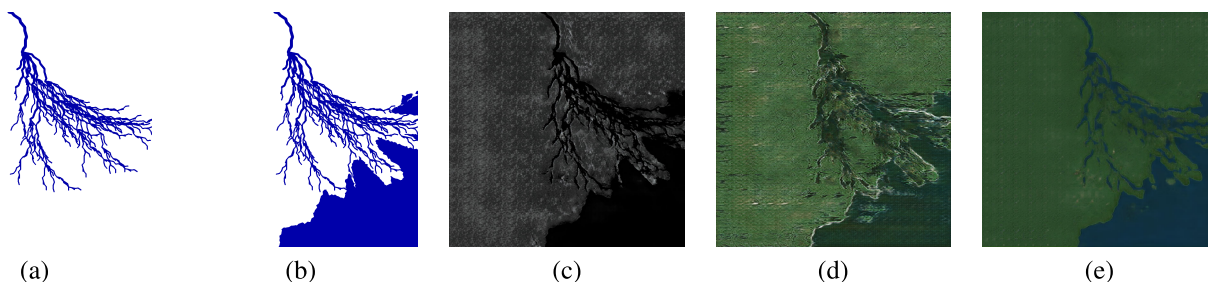


FIGURE 15. Another example of a delta generated by the first L-system of Table 3. (a) shows its graphical result. (b) The water map. (c) The height map generated by the first cGAN, while (d) and (e) show the polar and tropical climate images, respectively.

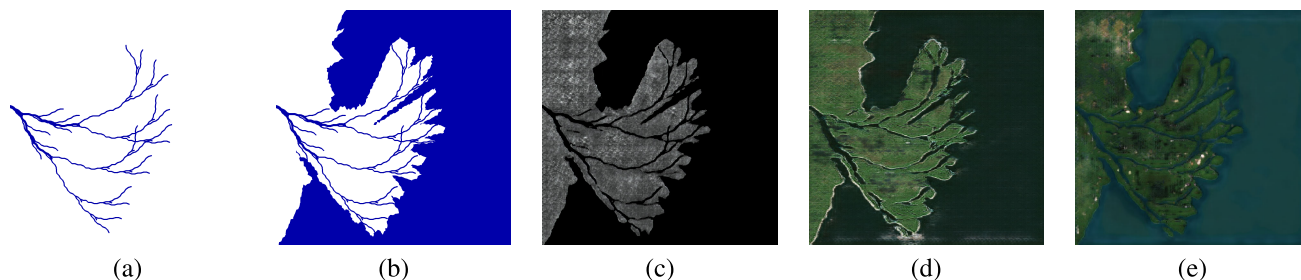


FIGURE 16. A delta generated by the second L-system in Table 4. (a) shows its graphical result and (b) its corresponding water map. (c) The height map generated by the first cGAN, while (d) and (e) show the polar and tropical climate images respectively, these were generated by independent cGANs using the height map as input.

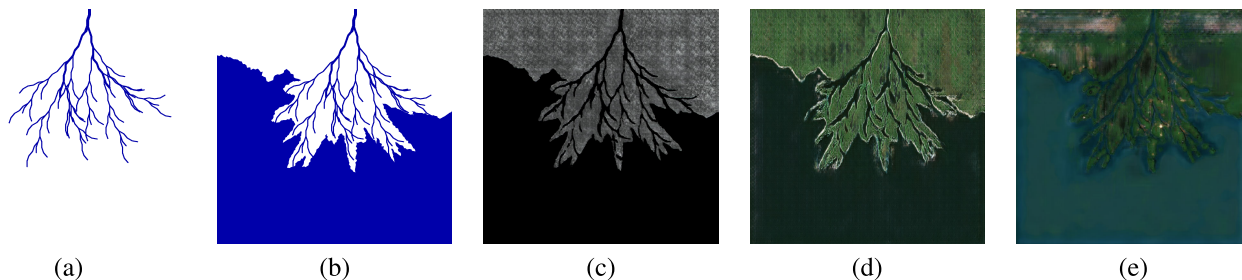


FIGURE 17. Another delta generated by the second L-system in Table 4. Column (a) shows graphical result, which was rotated 90°. While column (b) has its corresponding water map. The column (c) shows the height map generated by the first cGAN and columns (d) and (e) show the polar and tropical climate images respectively, these were generated by independent cGANs using the height map as input.

1) QUANTITATIVE EVALUATION

The evaluation of the image quality includes two metrics: the Fréchet Inception Distance (FID) [32] and the Naturalness Image Quality Evaluator (NIQE) [33]. FID measures

the similarity of GAN-generated images with real images, while NIQE is a no-reference evaluator. To compare the generated deltas against real images, we use the Full-Delta set of images of tropical and polar climates of the DRCA2020

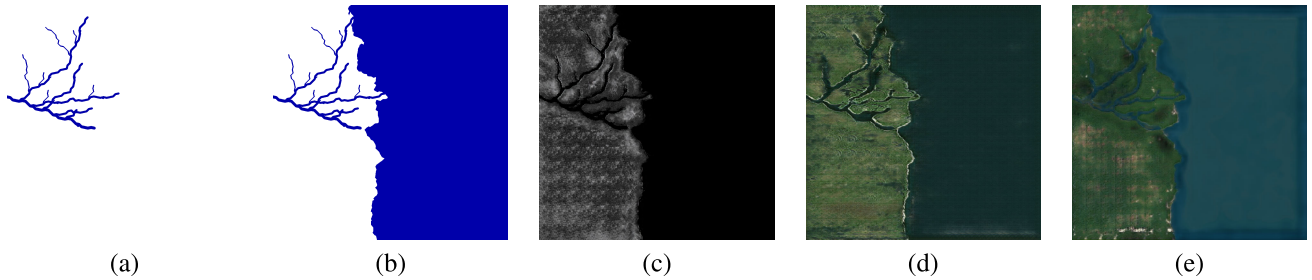


FIGURE 18. A delta resulting from the third L-system of Table 5. (a) Its graphical result and (b) its corresponding water map. (c) shows the height map generated by the first cGAN and (d) and (e) show the polar and tropical climate images.

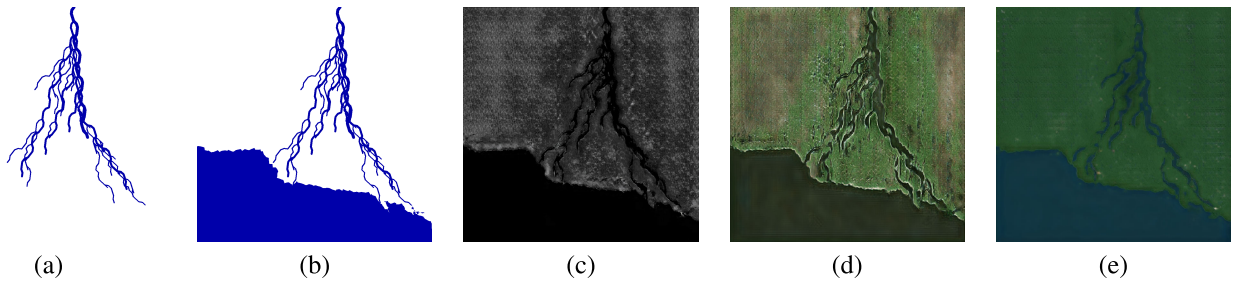


FIGURE 19. A different delta resulting from the third L-system shown in Table 5. (a) shows its graphical result; (b) the water map; (c) the height map generated. (d) and (e) show the polar and tropical climate images respectively, these were generated by independent cGANs using the height map as input.

dataset (available in <https://github.com/DRCA2020/Full-Deltas>). The Full-Deltas dataset has 22 images of the polar climate and 105 of tropical climate. The FID is a metric that shows how close a group of images is to another; comparing an image group with itself gives a distance of 0, despite the order of the images. The evaluation of the generated images was between two mutually exclusive groups of real images of the same climate. Thus, the real deltas were randomly divided into control and test groups (50/50%). The first comparison was between the control and test groups to determine a reference distance. The second comparison was between the control group and the processed test group; the processing for the test group added Gaussian noise with a mean of zero and a standard deviation of 0.3. This second comparison aims to establish the other reference interval, where an image has an inferior quality. Finally, the third comparison was between the control and 30 images generated by our modular method. Examples of each group are shown in Figure 20.

The quality evaluation with the NIQE uses the statistical features of a natural scene statistics model, which is focused on the image quality [33]. Therefore, in this quality evaluation, we used the original pre-trained model of NIQE, implemented in Matlab. We compared each image in the three groups (the control images, the processed test images and the generated images) obtaining the average divergence for each group. Again, we performed this evaluation for tropical and polar images.

The results of both the FID and NIQE evaluations are reported in Table 6 and Table 7. The tropical deltas from the DRCA2020 dataset have a FID of 8.86. This value is the

TABLE 6. Quality measurements for the tropical images.

	Control image set (ref. lower bound)	Generated by our method	Noisy image set (ref. upper bound)
FID	8.86	27.60	44.45
NIQE	2.93	3.66	29.72

TABLE 7. Quality measurements for the polar images.

	Control image set (ref. lower bound)	Generated by our method	Noisy image set (ref. upper bound)
FID	16.72	36.90	38.21
NIQE	4.63	9.18	29.94

lower reference distance for tropical deltas. The comparison between the control group and the generated images show a distance of 27.6, this distance is approximately in the middle of our reference interval, as the FID in the comparison between the control group and the noisy group is of 44.45. When measuring the NIQE of the images, in average the control group has a divergence of 2.93 from the model and our generated images come very close with an average of 3.66, while the noisy images have an evaluation of 29.27.

Regarding polar images, the FID of the generated images is closer to the FID of the noisy images; this could be because some features are missing in the generated images, such as the lakes which are common in polar climates. However, when measuring the quality of the images with the NIQE, the generated images are closer in quality to the real ones than to the noisy images.

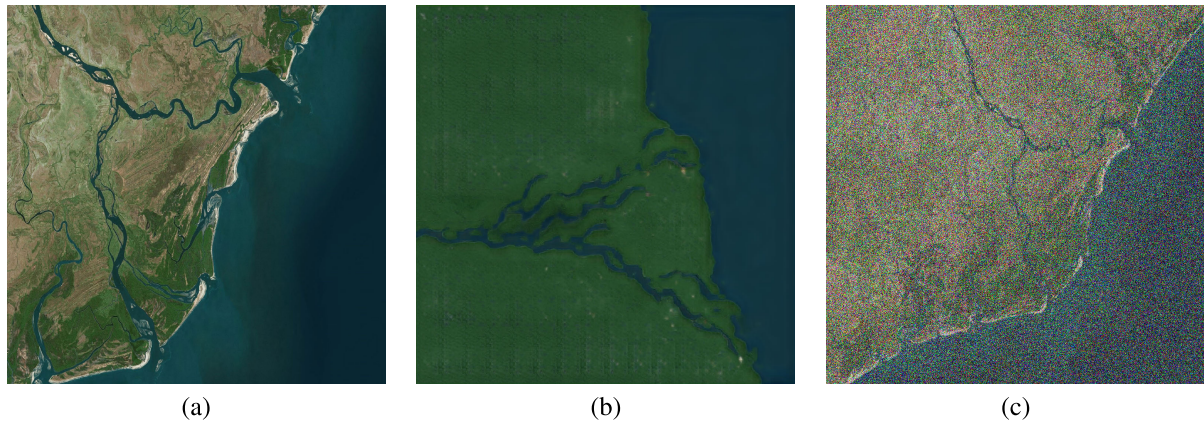


FIGURE 20. An example of images in the quantitative evaluation. (a) A real image of tropical climate in the control group. (b) A generated image using our proposed modular method. (c) A processed image of the test group decreased in quality.

2) EXAMPLE OF RELATED WORKS

There are a few methods for the automatic generation of river deltas. One of them is the algorithm proposed in [9], which converts a river mouth into a delta by generating an irregular semi-circle of new land linked to the former mouth to form the delta channels. Unlike our proposal, that method cannot generate complex deltas, as it only allows to have a single branching point. As a result, it can only generate simple river-dominated deltas. The delta generated in [9] is shown in Figure 21.

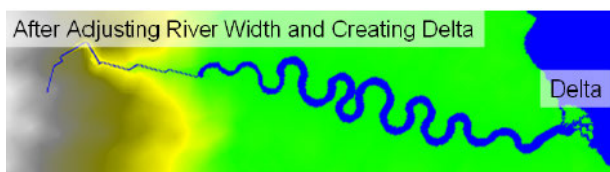


FIGURE 21. A delta generated by the method of Teoh, presented in [9].

Another way of generating deltas is with simulation methods such as in [7], which simulates the sediment transport process representing the terrain as a graph. At each step, the sediment movement is calculated, and the river channels are created as a result. Figure 22 shows the state of the delta after 1.2 million, 2.5 million and 5 million time steps in the sub-figures (a), (b) and (c) respectively. In contrast, our proposed approach generates four different images in each instance: a water map, its DEM and two different surface images. To the authors' knowledge there are no mature methods generating river deltas this way in order to perform a direct comparison.

3) INFERENCE TIME

The inference time of our modular approach was measured in a device with an Intel Core i7 (4th Gen) 4510U 2.0 GHz processor, 4 GB of RAM, and an Nvidia GTX 860m 2GB GPU. The average inference time for the generation of the images is presented in Table 8. An input water map is created in less than 2 seconds on average, and the three resulting



FIGURE 22. Image of a simulation method proposed in [7], at 1.2 million steps (a), 2.5 million steps (b), and 5 million steps (c).

images (DEM, tropical and polar landscape) are obtained in 22.5 seconds on average.

TABLE 8. Average time that each module takes to complete its generation task.

Module	Inference Time
Delta skeleton generation	0.5 seconds
Water map generation	1.2 seconds
Digital elevation model generation	7.5 seconds
Surface image generation	7.5 seconds

4) LIMITATIONS

The proposed approach has limitations. First, it can not generate seasonal water areas in the water map, producing noise in the land areas of the generated landscapes. Second, the channels created through stochastic L-systems can only grow into the same general direction. Deltas switch the growth to other areas when the slope is changed by the sediment deposited in the original area. This results in some channels that stop receiving the same amount of water and others that start receiving more. Third, although the guidelines help to generate useful deltas, a small number of them could be unrealistic. Therefore, the user must follow strictly the proposed guidelines to change variable parameters or between rule interactions. Thus, these shortcomings open up space for future research.

V. CONCLUSION

Although river deltas in nature are dynamic systems with complex features, our modular approach can generate

terrains with realistic river deltas structures. The first module proposes and follows a set of guidelines for rule creation and parameter adjustment for stochastic L-systems designing. The proposed design provides variability in creating original river delta skeletons. A second module processes the delta skeleton using a designed algorithm to create water maps. The construction parameters were empirically adjusted, allowing us to propose values that work with all the skeletons generated through our L-system. Through these parameters adjustment, it is possible to generate both river-dominated and tide-dominated deltas. Finally, integrating cGANs architectures in the third module produces elevation models and surface images of the generated deltas in polar and tropical climates. Therefore, the proposed modular approach introduces an automatic process to generate original and realistic river deltas. In the quality validation of the proposed method, the *Frechet Inception Distance* showed that the generated images are inside the control intervals. Thus, the generated images with our modular approach are similar enough to real images. Additionally, the *Naturalness Image Quality Evaluator* validates the effectiveness of this proposed method with values almost similar to the real images in the DRCA2020 dataset.

REFERENCES

- N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games* (Computational Synthesis and Creative Systems), 1st ed. Cham, Switzerland: Springer, 2016.
- G. Lai, W. Latham, and F. F. Leymarie, "Towards friendly mixed initiative procedural content generation: Three pillars of industry," in *Proc. Int. Conf. Found. Digit. Games (FDG)*, New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–4.
- R. J. Vitacion and L. Liu, "Procedural generation of 3D planetary-scale terrains," in *Proc. IEEE Int. Conf. Space Mission Challenges Inf. Technol. (SMC-IT)*, Jul. 2019, pp. 70–77.
- V. Sandfort, K. Yan, P. J. Pickhardt, and R. M. Summers, "Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks," *Sci. Rep.*, vol. 9, no. 1, pp. 1–9, Dec. 2019.
- J. Runge et al., "Inferring causation from time series in earth system sciences," *Nature Commun.*, vol. 10, no. 1, pp. 1–13, Dec. 2019.
- R. J. Hugget, *Fundamentals of Geomorphology* (Fundamental of Physical Geography), 4th ed. Evanston, IL, USA: Routledge, 2016.
- 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.
- Microsoft. *Bing World Map*. Accessed: May 10, 2020. [Online]. Available: <https://www.bing.com/maps>
- S. T. Teoh, "River and coastal action in automatic terrain generation," in *Proc. Int. Conf. Comput. Graph. Virtual Reality*, H. R. Arabnia and L. Deligiannidis, Eds. Las Vegas, NV, USA: CSREA Press, 2008, pp. 3–9.
- P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants* (The Virtual Laboratory). New York, NY, USA: Springer, 1996.
- L. O. Valencia-Rosado and O. Starostenko, "An approach for river delta generation using L-systems," in *Technology, Science and Culture—A Global Vision*, S. Picazo-Vela and L. R. Hernández, Eds., vol. 2, 2020, pp. 123–134.
- L. O. Valencia-Rosado, Z. J. Guzman-Zavaleta, and O. Starostenko, "Generation of synthetic elevation models and realistic surface images of river deltas and coastal terrains using cGANs," *IEEE Access*, vol. 9, pp. 2975–2985, 2021.
- 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.
- European Space Agency. *Global Surface Water Explorer*. Accessed: Aug. 15, 2019. [Online]. Available: <https://global-surface-water.appspot.com/map>
- I. Marák, B. Beneš, and P. Slavík, "Terrain erosion model based on rewriting of matrices," in *Proc. 5th Int. Conf. Central Eur. Comput. Graph. Vis.*, Plzeň, Czech Republic, Feb. 1997, pp. 341–350. [Online]. Available: <http://dspace5.zcu.cz/handle/11025/15907>
- D. Leitner, S. Klepsch, G. Bodner, and A. Schnepf, "A dynamic root system growth model based on L-systems: Tropisms and coupling to nutrient uptake from soil," *Plant Soil*, vol. 332, nos. 1–2, pp. 177–192, Jul. 2010.
- Y. I. H. Parish and P. Müller, "Procedural modeling of cities," in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*. New York, NY, USA: Association for Computing Machinery, 2001, pp. 301–308.
- G. S. Hornby and J. B. Pollack, "Evolving L-systems to generate virtual creatures," *Comput. Graph.*, vol. 25, no. 6, pp. 1041–1048, Dec. 2001.
- J. Togelius, N. Shaker, and J. Dormans, "Grammars and L-systems with applications to vegetation and levels," in *Procedural Content Generation in Games* (Computational Synthesis and Creative Systems), N. Shaker, J. Togelius, and M. J. Nelson, Eds. Cham, Switzerland: Springer, 2016, pp. 73–98.
- H. F. Yin and C. W. Zheng, "A practical terrain generation method using sketch map and simple parameters," *IEICE Trans. Inf. Syst.*, vol. E96.D, no. 8, pp. 1836–1844, 2013.
- K. Choros and J. Topolski, "Parameterized and dynamic generation of an infinite virtual terrain with various biomes using extended Voronoi diagram," *J. Universal Comput. Sci.*, vol. 22, no. 6, pp. 836–855, 2016.
- J. Doran and I. Parberry, "Controlled procedural terrain generation using software agents," *IEEE Trans. Comput. Intell. AI Games*, vol. 2, no. 2, pp. 111–119, Jun. 2010.
- A. Emilien, P. Poulin, M.-P. Cani, and U. Vimont, "Interactive procedural modeling of coherent waterfall scenes," *Comput. Graph. Forum*, vol. 34, no. 6, pp. 22–35, Sep. 2015.
- S. Jeschke, C. Hafner, N. Chentanez, M. Macklin, M. Müller-Fischer, and C. Wojtan, "Making procedural water waves boundary-aware," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation (SCA)*. Goslar, Germany: Eurographics Association, 2020, pp. 1–8. [Online]. Available: <https://doi-org.udlap.idm.oclc.org/10.1111/cgf.14100>
- J. Langr and V. Bok, *GANs in Action*. Shelter Island, NY, USA: Manning, 2019.
- 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*, vol. 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2014, pp. 2672–2680. [Online]. Available: <http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf>
- 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)*, vol. 1. Los Alamitos, CA, USA: IEEE Computer Society, Jul. 2017, pp. 1125–1134.
- X. Ning, D. Gou, X. Dong, W. Tian, L. Yu, and C. Wang, "Conditional generative adversarial networks based on the principle of homology continuity for face aging," *Concurrency Comput., Pract. Exper.*, Apr. 2020, Art. no. e5792. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1002/cpe.5792>
- 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.
- E. Nesvold and T. Mukerji, "Geomodeling using generative adversarial networks and a database of satellite imagery of modern river deltas," *Petroleum Geostatistics*, vol. 2019, no. 1, pp. 1–5, Sep. 2019. [Online]. Available: <https://www.earthdoc.org/content/papers/10.3997/2214-4609.201902196>
- É. Guérin, J. Digne, É. 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, pp. 1–13, Nov. 2017.
- M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, G. Klambauer, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a Nash equilibrium," 2017, *arXiv:1706.08500*.
- A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a 'completely blind' image quality analyzer," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 209–212, Mar. 2012.



LUIS OSWALDO VALENCIA-ROSADO was born in Campeche, Mexico, in 1989. He received the B.S. degree in international business management, the M.Sc. degree in industrial engineering, and the Ph.D. degree in intelligent systems from the Universidad de las Américas Puebla, in 2013, 2015, and 2021, respectively. His current research interests include procedural generation, generative models, and their applications in earth sciences and video games.



OLEG STAROSTENKO (Member, IEEE) was born in Lviv, Ukraine, in 1959. He received the B.S. and M.S. degrees in computer science from Lviv State University, Ukraine, in 1982, and the Ph.D. degree in mathematics and physics from the Autonomous University of Barcelona, 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 interests 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).

...



ZOBEIDA J. GUZMAN-ZAVALETA received the bachelor's degree 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 has worked as a Visiting Researcher at INAOE, in 2017, and a Visiting Professor at IBERO, Puebla, from 2016 to 2018, and CINVESTAV, Tamaulipas, Mexico, in 2018.

Currently, she has been an Associate Professor at the Universidad de las Américas Puebla (UDLAP), since 2019. She is a Mexican Computer Scientist interested in finding relevant and abstract information from complex data, such as video, images, and audio. Additionally, she is a member of the National Council of Researchers in Mexico (SNI) and the Asociación Mexicana de Computación (AMEXCOMP).