

Received August 17, 2016, accepted September 9, 2016, date of publication September 27, 2016,  
date of current version October 31, 2016.

Digital Object Identifier 10.1109/ACCESS.2016.2612700

# Synthetic Modeling Method for Large Scale Terrain Based on Hydrology

HUIJIE ZHANG<sup>1</sup> (Member, IEEE), DEZHAN QU<sup>1</sup>, YAFANG HOU<sup>1</sup>, FUJIAN GAO<sup>1</sup>,  
AND FANG HUANG<sup>2</sup>

<sup>1</sup>School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China

<sup>2</sup>School of Geographical Science, Northeast Normal University, Changchun 130117, China

Corresponding author: F. Huang (huangf835@nenu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 41671379, in part by the Natural Science Foundation of Jilin Province under Grant 20140101179JC, in part by the National Natural Science Foundation of China for Young Scholars under Grant 41101434, in part by the Research Fund for the Doctoral Program of Higher Education of China under Grant 20130043110016, and in part by the National Natural Science Foundation of China under Grant 41571405.

**ABSTRACT** Generating large scale terrains that conform to the morphology of real scenes is a great challenge for terrain modeling, as simulating complex geometric details is time-consuming and the realistic geographical features are hard to be controlled. In this paper, we propose an efficient modeling method for large scale terrain visualization based on hydrology. To simulate real geographic features, we introduce the hydrology-based Tokunaga river network to guide the terrain generation, and propose a production rule set of river network using procedural modeling. The distribution and structure of river network can be adjusted by user interactions. Ridges are extracted based on river network to provide more skeleton features, and the enrichment method of skeleton features is presented to maintain the morphology of valleys and ridges. Based on the enriched features, diffusion equation is exploited to compute the full elevation field, which can achieve the nature transitions of the regions between skeleton features. Large scale terrain with real morphological features can be generated online through the parallel implementation of diffusion equation. According to user requirements, the augmented virtual terrain can be obtained by blending the selected real terrain with the synthesis terrain seamlessly. The experiments are conducted on digital elevation model, and the results show that the proposed methods can generate large scale terrains that conform to the morphology of real terrain and can well simulate various natural scenes.

**INDEX TERMS** Terrain visualization, hydrology, river network, procedural modeling, augmented virtual terrain.

## I. INTRODUCTION

Virtual scenes have been widely used on fields of medicine and healthcare, geological analysis, video games and movies. Among all parts of a virtual scene, virtual terrain is one of the main components, and is a dominant visual element. In recent years, a great progress in terrain modeling has been made by many researchers, and many effective terrain modeling methods have been proposed. Existed techniques for terrain modeling can be classified as follows: procedural modeling methods [1], [2], modeling methods based on sketches [3], [4] and examples [5], and modeling methods based on physics [6]. Procedural modeling methods are usually efficient for generating large scale terrain, but they usually don't support good control of terrain morphology because of the randomness. Modeling methods based on sketches can generate terrain according to the morphology denoted by user

sketching. However, manual editing is usually inefficient and requires the professional knowledge of users. For methods based on terrain examples, features of real terrain can be reserved in the synthetic terrain by jointing the real terrain. But the efficiency of terrain piecing is low, which is hard to generate large scale terrain. Virtual terrains generated by methods based on physics algorithm can well conform to the morphology of real terrain. However, this kind of method is also hard to control and can't support the generation of large scale terrains. So the lack of controllability is a general issue for most of current methods. Moreover, how to efficiently generate large scale terrains that fit in the requirements of geology is a great challenge.

In this paper, we propose a procedural modeling method based on hydrology theory to generate virtual terrains that accord with the morphology of real terrain. Our system allows

users to control the terrain morphology interactively through easy-to-use operations. According to hydrology theory, river networks play important roles in terrain formation and are effective to ensure the real terrain morphology. So, we introduce fractal theory to generate river network, according to the self-similarity property of river network. Based on the Tokunaga river network [7], [8], we propose a set of production rules that make our river network accord with hydrology theory. Based on the river network, we extract the watersheds as ridge features, to constitute the whole skeleton features of terrain. The elevations of these skeletons are computed according to the distances from rivers to the estuary. To enrich the features of mountains and valleys, Midpoint Displacement method and its inverse process are exploited to extend the skeleton features, which can avoid that the features of mountains and valleys are too steep. On the basis of extended skeleton features, we exploit diffusion equation to compute the elevations of other regions to obtain the elevation field of the whole terrain. Moreover, we provide a terrain editing operation to embed a block of real terrain into the virtual terrain. For example, if users are interested in a block of real terrain, the augmented virtual terrain with features of real terrain can be obtained, through the terrain editing operation. Additionally, the performance for generating large scale terrain is enhanced, by the parallel implementation of elevation computation on GPU. The main contributions of our work are threefold:

1. We propose a production rule set based on Tokunaga self-similar network for generating river network that conforms to hydrology theory. The terrain generated based on our river network meets the accuracy requirements of scene modeling and supports the observations in the field of river science.

2. We propose a terrain modeling framework to efficiently generate large scale terrain with good controllability, combining procedural modeling with skeleton features.

3. We propose a terrain blending method based on Poisson equation to blend virtual terrain and real terrain seamlessly. Users can embed a block of real terrain into an arbitrary position of the generated terrain, which ensures that the features in real terrain are blended into the virtual terrain and the boundaries of real terrain are smoothly connected with the original virtual terrain.

The rest of this paper is organized as follows: we first review the related researches in Section 2. Then, the overview of our methods is described in Section 3. The details of our methods are demonstrated in Section 4, Section 5 and Section 6. The results of our method are presented in Section 7. The performance is discussed in Section 8, followed by the conclusion and future work in Section 9.

## II. RELATED WORK

### A. PROCEDURAL MODELING METHODS

Procedural modeling is a classical parametric method that is easy to operate and is suitable for generating large scale terrain. A detailed overview of procedural modeling method

was given by Ebert *et al.* [9]. Among the approaches of procedural modeling, the most representative one is the adaptive subdivision method, proposed by Fournier *et al.* [10], which provided a set of stochastic models to present various natural phenomena. Perlin [11] proposed a noise-based procedural method, which combined the noise function to present various details under different scales. Though fractal based approaches have been commonly used to generate details of large scale terrain, the procedural modeling methods lack the spatial information based on the topography features, which is hard to precisely present the morphological characteristics of landforms.

### B. TERRAIN MODELING BASED ON RIVER NETWORK AND SKETCH

Utilizing river network and sketch are two effective ways to generate terrains with specific morphology features. According to geology theory, real terrain is generally produced through erosion and weathering. So introducing the information of the river network to the terrain modeling process is a reasonable solution for generating realistic terrain [12]. Based on the Horton-Strahler river network classification method, Zhang *et al.* [13] concluded a description of Tokunaga fractal river networks on statistical average sense. Claps *et al.* [14] generated self-similar river networks through recursive and iterative operations. Researches on geological river network can provide morphological guidance for modeling realistic terrains. Terrains were generated through constructing river network in the work of Teoh [15], but the model of river network is not based on hydrology. Génevaux *et al.* [12] proposed a terrain generation method based on hydrology, using procedural models. They first modelled river network using a probabilistic way, and then constructed watersheds based on the river network. The final terrain was generated by blending river patches with the procedural terrain. The idea of their method is similar to ours. However, from the point of hydrology, our method introduces Tokunaga river network to simulate the real morphology of drainage basins, which better accords with hydrology theory. For terrain generation, we use diffuse equation to generate realistic landform under the guidance of river network and ridges, which can present the details of terrain more realistically. The context-sensitive L-system and Midpoint Displacement method were introduced by Prusinkiewicz and Hammel [16], to embed the morphology of rivers into terrain by using fractal modeling system. Kelley *et al.* [17] proposed a procedural method to generate river basins. Using this method, the local sketch can be produced as the input data of global terrain. However, only several abstract parameters can be used to control the river network, without providing user interactions. A fractal based algorithm was proposed by Belhadj and Audibert [18] to generate realistic landscapes. In their work, the terrain is constrained by the river network and ridge lines. However, the river network in this work was uncorrelated with information of the river basins. In 2014, according to perceptual cues, Tasse *et al.* [4] have found the best match between profile

sketches and the input terrain profile of ridges. In their work, to ensure that the user-defined profiles were not occluded by other parts of terrain, deformation was exploited for terrain to completely match the sketches from a perspective view.

### C. TERRAIN MODELING METHODS BASED ON PHYSICS

This type of methods usually exploits physical models to simulate the process of terrain generation. In view of the fact that the morphology of land surface is affected by multiple environmental factors, such as water erosion and change of temperature, modeling method based on physics was proposed by Musgrave *et al.* [19], who introduced a simple physical erosion model to simulate the details on land surface. By using different erosion algorithm, mountain sceneries were well simulated in the work of Chiba *et al.* [20]. An erosion model that provided high level of control has been presented in the work of [21], and a hydraulic erosion model inspired by fluid mechanics was deduced in the work of [22]. Wojtan *et al.* [23] introduced a morphology algorithm of corrosion simulation, which supports user interactions to control the shapes of objects. Krištof *et al.* [24] introduced smooth particle hydrodynamics to the erosion method. The algorithm efficiency is a nonnegligible issue for this kind of methods. Although GPU implementation can settle this matter to a large extent [25], [26], it is still hard to generate large scale terrain, because of the algorithm complexity.

### D. CONTROLLABILITY AND INTERACTIVITY OF TERRAIN MODELING

The controllability and interactivity are important for generating terrains that satisfy the user requirements. In order to better control the terrain construction, modeling methods with interactive editing have been studied by many researchers. Hnaidi *et al.* [27] described an algorithm to edit the river trajectories directly. In this method, terrain was generated based on a given set of topographic features, using multi-grid diffusion equation and gradient constraint for matching the corresponding control curve. Although this method is based on the differential properties of grid, and can generate large high-resolution terrain, the problem of physiognomy representation in rivers modeling is still existed. In order to solve this problem, Skytt *et al.* [28] exploited the novel LR B-splines [29] in the processing of the huge geographical data. Surfaces modeled by LR B-splines can provide the adjustment mechanism of local details and the compact representation of global data. Rusnell *et al.* [30] proposed a feature based synthetic modeling method to add controllability of terrain morphology. According to the sketch features, Zhou *et al.* [31] generated the final terrain based on the synthesis of example terrains. Although very well results have been achieved in this work, an input terrain must be given in advance, which is also the common problem of terrain synthesis methods based on example terrains. Under the constraints of user sketches, Gain *et al.* [32] utilized multiresolution surface deformation to generate terrains that satisfied user requirements. However, incorrect results may

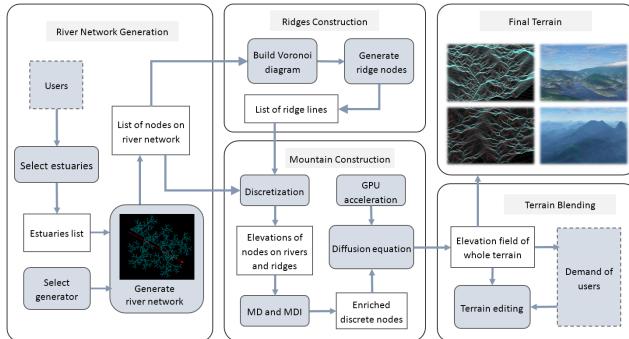
be produced from a geologic perspective if the users don't have professional knowledge. To address this issue, some researchers have combined erosion and deposition algorithms with interactive editing [6]. But this kind of method cannot support the generation of large scale terrains. In this paper, we provide flexible user interactions to control the start position and grow direction of the river network in a large scale scene. Furthermore, the morphology of rivers and mountains is enriched by Midpoint Displacement method and its inverse process.

## III. OVERVIEW

In this paper, we propose a terrain synthetic modeling method to generate realistic terrain that conforms to the geographical morphology through a simple and efficient process with high user controllability. To make the generated terrain accord with the morphology of real terrain, we create self-similar river network based on hydrology theory, as the base of terrain modeling. The height values of the river network are determined by the distances from rivers to estuary. To contain more abundant features in our generated terrain, we build Voronoi diagram based on the river network and extract ridges to form the whole skeleton features. Discretization of the skeleton features is performed by Bresenham algorithm. Then the elevations of ridges are calculated according to the heights of river network. To generate the whole terrain based on skeleton features, the morphology of mountains and valleys should be further described. So Midpoint Displacement (MD) and Midpoint Displacement Inverse (MDI) are utilized to generate more data points for enriching the morphology features of river network and ridges. After obtaining elevations of enriched skeleton features, the whole elevation field of the generated terrain is computed through diffusion equation based on Laplace. At last, we provide a flexible terrain editing operation based on Poisson equation to integrate existed real terrain and the generated virtual terrain. The user controllability contains two aspects, including drawing vector lines for controlling the generation of river network and selecting terrain features for terrain blending. The schematic representation of our workflow is shown in Fig. 1.

## IV. RIVER NETWORK CONSTRUCTION

In this section, we describe the details of the river network construction based on hydrology theory. Horton-Strahler river network classification is a classical method that exploits multiple measures to quantitatively describe the river network [33]. Through introducing the fractal theory, researchers have described the topology of self-similar network (SSN) through links. Tokunaga river network [7], [8] is a kind of self-similar river networks based on the Horton-Strahler river network classification. It can satisfy the plane-filling property of river network by using its fractal rules and can well describe the morphology of real river network. Zhang *et al.* [13] have given a basic generator sequence of SSN and deduced the parameter of Tokunaga river network based on this generator sequence. They found that the



**FIGURE 1.** Workflow of our method. The framework is composed of the generation of river work, construction of ridges, construction of mountain and terrain blending. Final terrain can simulate morphology of real terrain scenes and can be edited by users according to the requirements.

parameter can well describe the morphology of river network. In this paper, we propose a set of fractal rules to generate river networks that are consistent with real natural landscape, through combining the basic generator sequence with the parameter of Tokunaga river network. The generated river network is the base of our terrain modeling method.

**TABLE 1.** Relationship of  $\varphi$  and  $R$ .

$\varphi$	1	2	3	4
Generators				
$P$	--	1.00	0.768	0.640
$R$	--	1.00	1.535	1.921

#### A. QUANTIZATION OF RIVER NETWORK MORPHOLOGY

Self-similar river network is composed of a set of basic generators. Based on specific fractal rules, river network is generated from the initial generator through replacing the links with generators iteratively. Since most of river networks are formed as tree structure, we define generators of our river network based on binary tree. The generators of river network are made up by interior links and exterior links that respectively describe the main streams and tributaries. Similar to the work by Zhang et al. [13], we also adopt an asymmetric structure for our basic generators. Furthermore, we assign different angles to exterior links for distinguishing the tributaries in different phases. Four fractal generators with different numbers of links are shown in Table 1, in which the arrow orientation points out the direction of debouchure, the vertical links denote the interior links, and the non-normal links denote the exterior links. We let the two exterior links farthest from debouchure rotate  $\theta$  angle about interior links, and let other exterior links rotate  $2\theta$  angles about interior links, which better conforms the morphology of real river

network through practical observations. Based on the basic characteristic of binary tree, we can obtain that the numbers of exterior links are one more than the number of interior links. To quantize the description of river network, we arrange the basic generators as a sequence in the ascending order of link numbers. Thereby, the growth pattern of river network can be described numerically by the generator index  $\varphi$ . After determining the generator sequence, we use Tokunaga fractal rules to generate the self-similar river network. Tokunaga fractal rules ensure that the growth of exterior links is prior to interior links, which conforms to the growth rules of real river network. So the exterior links are replaced by the generators with high serial numbers, whereas the interior links are replaced by the generators with relatively low serial numbers. Besides, the plane-filling property can also be satisfied by combining the generators with Tokunaga fractal rules according to the verification by Zhang et al. [13], which is essential for river network modeling. Given the parameter  $\varphi$  that denotes the serial numbers of the initial generator, we use generator  $\varphi$  to replace exterior links and replace interior links with generator  $R$  that is computed as follow:

$$R = P * (\varphi - 1), \quad (1)$$

in which  $P = \frac{1+\sqrt{1+4\varphi}}{2\varphi}$ . If  $R$  is not an integer, the decimal part of  $R$  is used as the probability to select a generator between its floor and ceil. For example, the value of  $R$  is 1.535 with  $\varphi = 3$  as shown in Table 1. In this condition, we have the probability of 53.5% to select generator 2 and the probability of 46.5% to select generator 1. The values of  $R$  for basic generators are presented in Table 1.

Besides, the lengths of links are same in one iteration step and are changed as the river network grows. They are computed as:

$$l_t = (1/P \cdot \varphi)^{t-1}, \quad (2)$$

in which  $t$  denotes the current iteration step,  $l_t$  is the link length in iteration step  $t$ . Through performing the fractal process iteratively, the self-similar river network that conforms to hydrology theory is generated.

#### B. PROCEDURAL MODELING FOR RIVER NETWORK BASED ON L-SYSTEM

Based on the quantization of river network, we propose a rule set to describe the evolution of river network through fractal L-system. Fractal L-system is a modeling technique to generate scenes by exploiting production rules, which can be embedded into algorithms. According to self-similar and fractal theory, we propose a production rule set based on the generators to describe the evolution of river network (interior links and exterior links). The basic operations in the rules are described in Table 2 and the rule set of river network generation are defined as follows:

*Definition 1:* Let  $F$  be a line segment that represent an interior link,  $G$  be a line segment that represents an exterior link, and  $D_{ep}$  be the maximum iteration depth.  $l$  is the

**TABLE 2.** Basic operations of generator links evolution.

symbol	operation
$\rightarrow$	a replacement operation during iteration
[	a push operation of location stack
]	a pop operation of location stack
+	rotate an angle clockwise
-	rotate an angle counter clockwise

length of line segments that are corresponding to  $F$  and  $G$ . Let  $F'$  be the interior links in the previous iteration step, and  $G'$  be the exterior links in the previous iteration step, then  $\Omega$  is the following rule set for river network generation in one iteration.

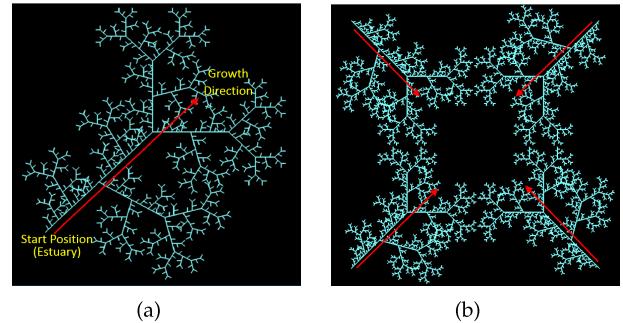
- I    For the case of  $\varphi = 2$ :  
Exterior links  $G' \rightarrow F[- - G]F[-G][+G]$ ;  
Interior links  $F' \rightarrow F[+ + G]F$  or  $F' \rightarrow FF$ .
- II    For the case of  $\varphi = 3$ :  
Exterior links  $G' \rightarrow F[+ + G]F[- - G]F[-G][+G]$ ;  
Interior links  $F' \rightarrow F[-G]F[-G][+G]$  or  $F' \rightarrow F[+ + G]F$  or  $F' \rightarrow FF$ .
- III    For the case of  $\varphi > 3$ :  
Exterior links  $G'$  are replaced by generator  $\varphi$ ;  
Interior links  $F'$  can be replaced by a generator according to the value of  $R$  or use the rule  $F' \rightarrow FF$ .

To ensure the generality and naturality of the produced river network, exterior links are replaced by generator  $\varphi$ , and interior links are substituted for one of the available generators according to a probability computed by the value of  $R$  or are lengthened directly.

The river network generated by rule set  $\Omega$  is shown in Fig. 2, in which the red vector lines are input by users to denote the start position and the growth direction of river network. Through setting the initial generator index  $\varphi$  and the maximum iteration depth  $Dep$ , the morphology of our generated river network can be controlled. Fig. 2(a) is the river network generated with  $\varphi = 2$  and  $MaxDepth = 7$ , which can describe the tree structure of real river system. As larger as the value of  $\varphi$ , the morphology of river network tends to be more featherlike. Conversely, the dendritic river network is more possible to be presented. As shown in Fig. 2(b), multiple river networks can be generated in one scene. The reliability of our generated river network is verified in Section 7, by the comparison with river network of real terrain extracted by ArcGIS.

### C. HEIGHT COMPUTATION OF RIVER NETWORK

To present the tendency of water flow, we compute the slopes of river segments according to the Geodesic distances from river network nodes to the estuary point. The elevation of river network in estuary point is set as 0. The heights of river network nodes are directly proportional to their Geodesic distances to the estuary point. As the fractal river network grows, heights of river network nodes are calculated step



**FIGURE 2.** River network produced by our generation rules. (a) River network generated by one user-input vector line with  $\varphi = 2$  and  $MaxDepth = 7$ . (b) River network generated by four user-input vector lines.

by step according to their distances to the estuary point. We provide an easy-to-use interactive tool to set the estuary point and general flow direction of river network, through users drawing straight line segments simply on a null height field. In this way, realistic river network with heights can be generated, by setting the start position and the growth direction of river network in L-system as the start point and orientation of the line segment. The iteration depth  $depth_L$  is computed as follow:

$$depth_L = length_{ab}/width_T \times MaxDepth, \quad (3)$$

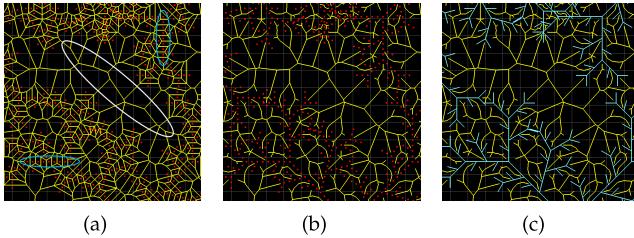
in which  $length_{ab}$ ,  $width_T$  and  $MaxDepth$  respectively denote the length of the line segment drawn by users, width of the terrain DEM and the given maximum iteration depth.

## V. TERRAIN GENERATION

In this section, to obtain more morphology features for modeling realistic terrain, we further produce ridges of terrain based on the generated river network. In geography, ridges are also important features for terrain generating. At the same time, for many real scenes, watersheds are corresponding to ridges. So the ridges can be obtained by extracting watersheds of the river network. Similar to the work in [12], we construct Voronoi diagram based on the nodes of our Tokunaga river network, and watersheds are the edges of Voronoi diagram among different river basins. Due to the continuous edges of skeleton features are difficult to be used in the grid data of DEM, we discretize the skeleton features onto the DEM using Bresenham algorithm. Furthermore, neighbor points of skeleton features are interpolated using Midpoint Displacement (MD) and its inverse process (Midpoint Displacement Inverse, MDI) [18] to ensure authentic shapes of rivers and mountains. Finally, diffusion equation based on Laplace is exploited for all the generated elevation points to compute the whole elevation field.

### A. CONSTRUCTION OF RIDGES

In the field of geographical information system, Voronoi diagram is commonly used for efficient neighbor interpolation and analysis of the influenced regions of geographic



**FIGURE 3.** Constructing ridges by building Voronoi diagram based on nodes of river network and cutting the edges crossed with the river network. Red points are the nodes of river network. Yellow lines are the edges of Voronoi diagram. Cyan line segments denote the river network. (a) Building Voronoi diagram based on nodes of river network. (b) Cutting the edges crossed with the river network. Watersheds separating different river basins are obtained. (c) Skeleton features composed of river network and ridges.

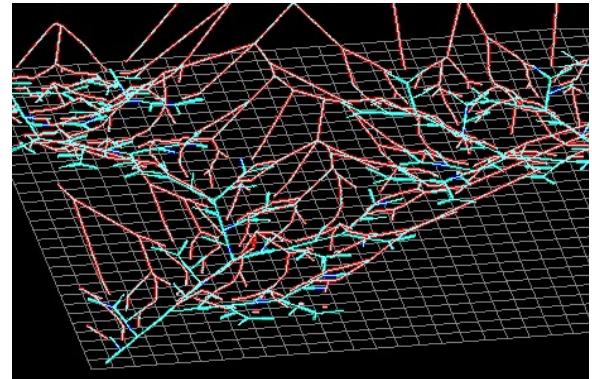
entities. Voronoi diagram satisfies the properties as follows: each Voronoi cell has only one point; and two adjacent points have the same distance to the edge between the two points. According to these properties, we construct Voronoi diagram based on the nodes of the river network using the classical Sweep Line algorithm [34], as shown in Fig. 3(a). The edges of Voronoi diagram can be processed differently according to whether the edges intersect with river network or not. For the edges that don't intersect with the river network, they can be considered as watersheds because these edges usually divide two different river basins. Therefore, we cut the edges that intersect with the river network, and preserve the uncrossed edges as ridge lines, as shown in Fig. 3(b). In Fig. 3(c), the ridge lines denoted by yellow lines and the river network denoted by cyan line segments constitute together the whole skeleton features of terrain. The generated ridges can well simulate the morphology of real terrain, which is demonstrated in Section 7.

For the heights of ridge nodes, we calculate them according to the heights of river network nodes as follows [12]:

$$Height_q = \max(Height_{p_1}, Height_{p_2}, Height_{p_3}) + \alpha \cdot d, \quad (4)$$

where  $q$  is a ridge node,  $p_1, p_2$  and  $p_3$  are the first three nearest nodes on river network to node  $q$ , and  $Height_q$ ,  $Height_{p_1}$ ,  $Height_{p_2}$  and  $Height_{p_3}$  are respectively the elevation values of nodes  $q$ ,  $p_1$ ,  $p_2$  and  $p_3$ . Besides,  $d$  represents the distance from  $q$  to its nearest river network node  $p_1$ , and  $\alpha$  is a parameter for adjusting heights. The 3D skeleton features of our virtual terrain are shown in Fig. 4. We can see that the river network is located around the ridges, which conforms to the cases of real scenes.

Terrain is usually represented by discrete data points. Skeleton features play an important role in presenting terrain morphology. So, for obtaining the whole terrain based on skeleton features, we discretize the segments of skeleton features onto grids, using classical Bresenham method. In order to make ridges better conform to the natural case, the heights of the discrete points are added by random values that



**FIGURE 4.** Produced 3D terrain skeleton composed of river network and ridges after the computation of heights. Cyan segments denote the river network with elevations. Red segments are the ridges.

accord with Perlin noises [11], and are computed as follows:

$$\begin{aligned} Height_P = Height_{P_s} + \frac{\|P - P_s\|}{\|P_e - P_s\|} (Height_{P_e} - Height_{P_s}) \\ + F(P), \end{aligned} \quad (5)$$

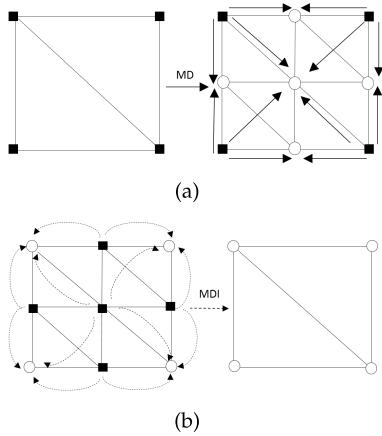
in which  $P$  is an interpolation point,  $P_s$  and  $P_e$  are respectively the start point and end point of a ridge line, and  $Height_P$  and  $Height_{P_s}$  and  $Height_{P_e}$  are respectively the elevation values of  $P$ ,  $P_s$  and  $P_e$ .  $F$  denotes the fractal Perlin noise function.

#### B. DATA EXTENSION OF SKELETON FEATURES

Based on the discrete skeleton features, the whole terrain can be computed through many methods. However, the terrain interpolated directly has too steep ridges and valleys. So, we extend the skeleton features to generate more feature points, exploiting Midpoint Displacement and its inverse process. Through this process, the morphology of mountains and rivers can be better presented and more realistic terrain is produced. Moreover, the whole elevation field can be generated more efficiently, because the elevations of more discrete points are known previously.

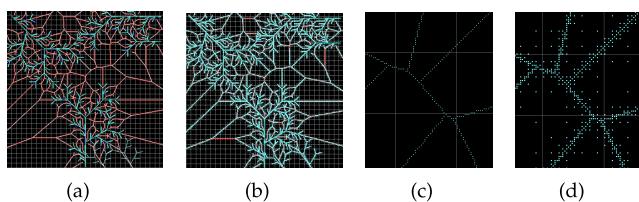
Midpoint Displacement (MD) is commonly used to generate fractal terrains. Through iterative dividing, terrain can be generated with the resolution required by users. The MD process can interpolate the square center and four midpoints of the edges based on four corner points. Midpoint Displacement Inverse (MDI) interpolates the corner points of a square based on the known data points inside the square. The process of MDI and MD are illustrated in Fig. 5. Through combining these two methods, skeleton features can be extended with dense neighbor points. Compared with linear interpolation, the extended mountains and rivers better conform to the morphology of real terrain.

Since the discrete data points of skeleton features are adjacently located on the grid with the required resolution, we first perform MDI process for these points to extend the regions of skeleton features. However, in the skeleton features region extended by MDI, some points inside square might be absent. So we next perform MD process to fill in the null points



**FIGURE 5.** Schematic diagram of MD and MDI methods. Black block dots denote the discrete points with computed elevation. White circular dots represent the null points need to be interpolated. (a) Midpoint displacement method. Directions of solid arrowed lines describe the interpolation process of MD. (b) Midpoint displacement inverse method. Directions of dashed arrowed lines illustrate the interpolating process of MDI.

inside the region of skeleton features. Through continually performing MDI process and MD process in turn, skeleton features can be significantly enriched. The more times the process performs, the more data points with elevations can be obtained. However, as the distance  $d$  between the two points becomes long, error of the interpolation using these two points becomes large. To address this issue, a parameter  $L$  is introduced to reduce the number of anomalous points in the interpolation. When  $d > L$ , we stop the interpolation. The results of feature enrichment are shown in Fig. 6 with resolution  $1024 \times 1024$ . Fig. 6(a) and Fig. 6(c) show the skeleton features before feature enrichment respectively in the global and detail view. From Fig. 6(b) and (d), we can see that dense points have been generated around the skeletons.



**FIGURE 6.** Enriching the skeleton features by processes of MD and MDI. Many discrete points are interpolated according to the heights of skeleton features. (a) Skeleton features without feature enrichment. Red lines are ridge lines. Cyan lines are river network. (b) Result of the enriched skeleton features from 2D perspective. Dense cyan points are interpolated based on skeleton features. (c) Details of the skeleton features before feature enrichment. (d) Details of the skeleton features after feature enrichment. More discrete points can be obtained.

### C. TERRAIN GENERATION BASED ON DIFFUSION EQUATION

After obtaining enough discrete data points of skeleton features, we generate the elevation field of the whole terrain, using diffusion equation based on Laplace. In the field of

physics, diffusion equation is exploited to study the temperature distribution for homogeneous objects with isotropic material, based on the heat transfer with environmental media. In this paper, we introduce the idea of diffusion equation to obtain the whole terrain data based on the difference in elevations. On the base of our feature model, the 3D skeleton features and non-null data points generated by MD and MDI are considered as the diffusion sources. Then, the elevations of null points can be calculated iteratively from the diffusion sources. The mathematical model of diffusion equation that describes the distribution of elevations at iteration step  $t$  can be demonstrated as follows:

$$\frac{\partial u(x, y, z, t)}{\partial t} = D\Delta u(x, y, z, t) + f(x, y, z, t), \quad (6)$$

in which  $u(x, y, z, t)$  represents the elevation on position  $(x, y, z)$  at iteration step  $t$ ,  $f(x, y, z, t)$  denotes the growth number of non-null points in current iteration step,  $\Delta$  is the Laplacian operator, and  $D$  is the diffusion coefficient. The Laplacian operator used in Equation (6) is

$$\Delta = 4U_{i,j} - U_{i+1,j} - U_{i-1,j} - U_{i,j-1} - U_{i,j+1}, \quad (7)$$

in which  $U_{i,j}$  is the elevation of a point with coordinate  $(i,j)$  on discrete grid, and  $U_{i+1,j}$ ,  $U_{i-1,j}$ ,  $U_{i,j-1}$  and  $U_{i,j+1}$  are elevations of the 4 adjacent points of point  $(i,j)$ . By performing the diffusion iteratively, the whole elevation field is obtained. Moreover, morphology of real terrain can also be ensured, because the diffusion process is constrained by the elevations of ridges and river network. Additionally, Perlin noises or Gaussian smoothing can be performed onto the generated terrain according to the requirements of users.

### VI. GENERATION OF AUGMENTED VIRTUAL TERRAIN

In order to satisfy the requirements of containing features of real terrain in the virtual terrain, we propose a terrain blending method based on Poisson equation. It allows users to embed a block of real terrain into our virtual terrain, and morphology features of real terrain can be reserved, with the seamless connection between real terrain and virtual terrain. To achieve this goal, we compute the gradient field of real terrain block to describe the terrain features and exploit Poisson equation to blend the terrains according to the elevations of the boundaries on the virtual terrain.

#### A. POISSON EQUATION

Poisson equation has been used in mesh processing to calculate the smoothly deformed mesh under the guidance of a gradient field [35]. In this paper, we utilize Poisson equation to blend our virtual terrain with a block of real terrain seamlessly, and take the block of real terrain, the block of virtual terrain and the block of blended terrain respectively as reference terrain  $R$ , source terrain  $S$  and target terrain  $T$ . These three terrain blocks have the same length  $len$ , the same width  $wid$  and the same points number  $m = len \times wid$ . Since the height field of DEM is a piecewise linear model, it can be considered as a discrete potential field. Meanwhile,  $S$  and  $R$

have the same topology, because the grid of DEM is uniform. So, Poisson equation is suitable for this paper to construct the augmented virtual terrain. For target terrain  $T$ , its boundaries are set as the boundaries of  $S$ , and the elevations of its inner regions can be obtained by solving Poisson equation based on the divergence field of  $R$  and boundaries of  $S$ . To obtain the unknown elevations in  $T$ , the divergence field  $\text{Div}$  of  $R$  is introduced in Poisson equation to describe the tendency of elevations in  $R$ . The divergence value  $\text{Div}_{i,j}$  of point with coordinate  $(i, j)$  can be computed based on the gradients of its neighbor points, as follows:

$$\begin{aligned} \text{Div}_{i,j} = & (\text{GarX}_{i,j-1} - \text{GarX}_{i,j+1})/2 \\ & + (\text{GarY}_{i+1,j} - \text{GarY}_{i-1,j})/2, \end{aligned} \quad (8)$$

in which  $\text{GarX}$  and  $\text{GarY}$  are respectively the gradient field of  $R$  in  $X$  direction and  $Y$  direction of the terrain plane.  $\text{GarX}_{i,j-1}$ ,  $\text{GarX}_{i,j+1}$ ,  $\text{GarY}_{i+1,j}$  and  $\text{GarY}_{i-1,j}$  are the gradient values of the neighbor points, which are computed by the values of elevation differences in the corresponding directions. These divergence values are formed as a vector  $b = (b_1, \dots, b_k, \dots, b_m)'$ , in which  $b_k = \text{Div}_{i,j}$ , ( $k = i \times \text{wid} + j$ ). Additionally, the boundaries of  $S$  are taken as the constraints to ensure the smooth connection between  $T$  and the original virtual terrain. For a point with coordinate  $(i, j)$  in  $T$ , if it has neighbor points that are located on the boundaries of  $S$ , the corresponding component  $b_k$  ( $k = i \times \text{wid} + j$ ) in vector  $b$  is further subtracted by the elevations of these boundary points in  $S$ . Thereby, vector  $b$  combines the divergence field of  $R$  with boundaries of  $S$ . Then, the discrete Poisson equation can be represented as the linear system

$$Ax = b, \quad (9)$$

in which  $A$  is a sparse coefficient matrix with size  $m \times m$ , and vector  $x = (x_1, \dots, x_k, \dots, x_m)'$ , ( $k = i \times \text{wid} + j$ ) denotes the unknown elevation values of  $T$ . In this paper, we adopt the Laplace operator described in Equation (7) in Poisson equation, so matrix  $A$  is the corresponding Laplace matrix. This means that the result of processing Laplace operator onto the unknown elevation values in  $T$  should conform to the divergence field of  $R$  and boundaries of  $S$ . Through solving the linear system, elevations of the data points in  $T$  can be obtained.

For the application of multi-resolution terrain, a mesh parameterization can be first performed. Thereby, the gradient fields can be computed for the parameterization discrete fields of each point. The work for multi-resolution case is left to future work.

## B. TERRAIN BLENDING

In this section, we describe the process of blending the virtual terrain and real terrain block. If users are interested in a specific feature in real terrain, they can select a terrain block containing this feature as reference terrain  $R$ . Then the divergence field of  $R$  is calculated to simulate the variation trend of the specific features. Through manipulating the divergence field

### Algorithm 1 Terrain Blending

---

**Input:** Reference terrain block  $R$  and source terrain block  $S$

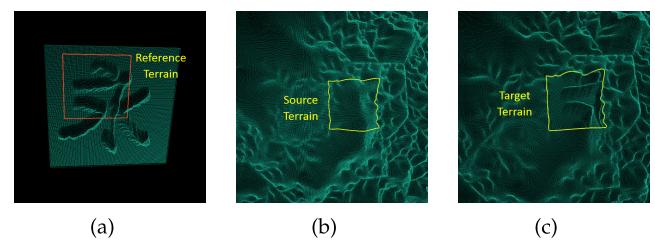
**Output:** Results of target terrain block  $T$

- 1: Compute the gradient field  $D$  of  $R$ ;
- 2: Users transform the gradient field  $D$  according to their requirements;
- 3: Set the boundaries  $B_s$  of  $S$ ;
- 4: Perform Poisson equation:
  - 4.1 Calculate the divergence field of  $R$  based on  $D$ ;
  - 4.2 Solve the linear system in Equation (9);
- 5: If the result of  $T$  can not satisfy requirements of users, go to Step 2;
- 6: Algorithm end.

---

of reference terrain, users can control the size of the features in  $R$ . Transformations such as scaling can be performed onto the gradient field. As the gradient field of reference terrain block is obtained, we choose a block region on the source terrain and blend the reference terrain and source terrain on this region. Algorithm 1 explained the process of terrain blending.

We take an artificial terrain with obvious features as the reference terrain to illustrate the effectiveness of terrain editing. As shown in Fig. 7(a), the reference terrain  $R$  selected by users is marked by the red square. The boundaries of source terrain  $S$  are marked by the yellow square in Fig. 7(b), and the target terrain  $T$  is shown in Fig. 7(c). We can see that the features of  $R$  in the red square of Fig. 7(a) are contained in the target terrain  $T$  marked by the yellow square in Fig. 7(c). Besides, the connection regions of two terrains are seamless, because the boundaries of source terrain block are used to ensure the smooth transition between two terrains.

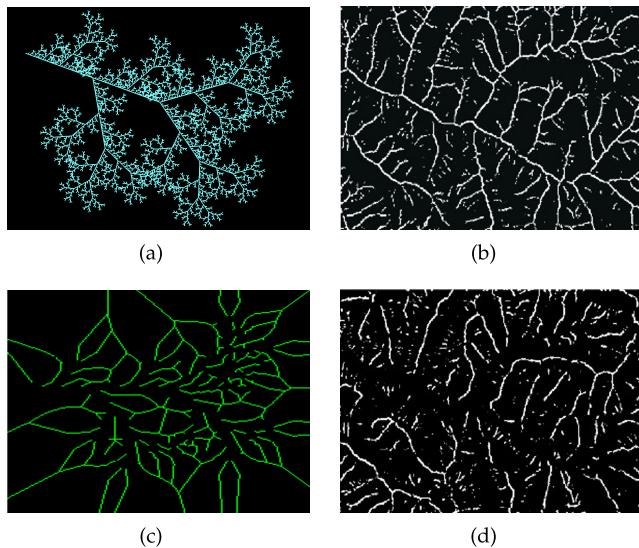


**FIGURE 7.** Results of terrain blending. Through solving Poisson equation, the wanted feature in another terrain can be blended into our virtual terrain. (a) An artificial terrain with features that are easy to be recognized. Red square denotes the source terrain. (b) Our virtual terrain. Yellow square denotes the target terrain. (c) Result of target terrain. The wanted features are embedded into our virtual terrain smoothly.

## VII. RESULTS

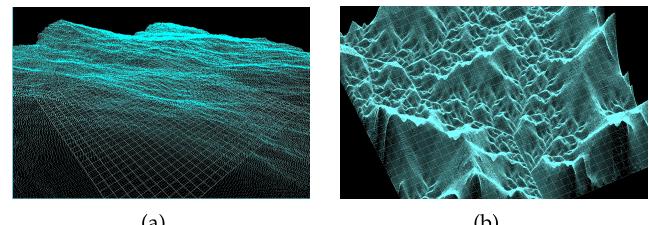
The system in this paper is developed using C++ to implement the algorithms. OpenGL Shading Language (GLSL) is utilized to render the terrain. Additionally, OpenCL is used to enhance the performance of terrain diffusion, by using parallel resources on GPU.

ArcGIS is a popular GIS software that can extract ridge lines and valley lines accurately. To verify the reliability of the river network and ridges generated by our methods, we compare our results with the valley lines and ridge lines extracted by ArcGIS. River networks in real scenes are usually presented as tree structure that has a main stream and some alternately arranged tributaries. River network generated by our method can also conform to this pattern. A typical example of valley lines in real terrain scenes are shown in Fig. 8(b), whose structure is very similar to binary tree. By comparing Fig. 8(a) and Fig. 8(b), we can find that the two river networks have similar patterns, because our production rules of river network conform to hydrology theory. Fig. 8(d) is the corresponding ridge lines of the terrain in Fig. 8(b). From Fig. 8(c) and Fig. 8(d), we can both find some continuous ridges with similar morphology, which actually separate different river basins. So, the skeleton features generated by our methods can well describe the skeleton features in real terrain scenes and provide a good base for generation of the whole terrain.

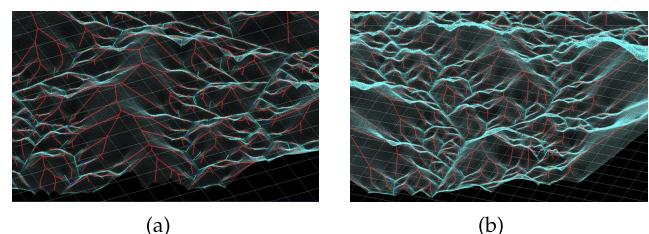


**FIGURE 8.** Comparison between results of our method and real feature lines extracted by ArcGIS. (a) Valley lines generated by our methods. (b) A typical example of valley lines in real terrain scenes extracted by ArcGIS. (c) Ridge lines generated by our methods. (d) Corresponding ridge lines extracted by ArcGIS.

To demonstrate the controllability of our method for terrain modeling, we compare our methods with the classical Midpoint Displacement method. As shown in Fig. 9(a), the terrain generated by Midpoint Displacement method present no specific geomorphic features, and users can only control the terrain generation by setting the elevations of four corner points. As shown in Fig. 9(b), the terrain generated by our method clearly present realistic geomorphological characteristics, such as riverways and watersheds, because our terrain is generated based on the skeleton features of rivers and mountains. Moreover, the skeleton features can be controlled by users through easy-to-use interactions and parameters setting.



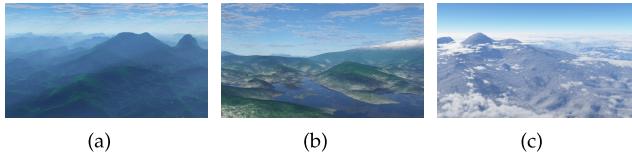
**FIGURE 9.** Comparing the terrain generated by our methods with the terrain generated by classical Midpoint Displacement. (a) Terrain generated by classical Midpoint Displacement method. Nearly no specific geomorphological characteristics can be presented. (b) Terrain generated by our methods. Riverways and mountains can be recognized.



**FIGURE 10.** Results of final terrain generated by our methods. Red segments are ridge lines. Cyan segments denote the river network. (a) Observing the terrain in the perspective that focuses on ridges. (b) Observing the terrain in the perspective that focuses on valleys.

The results of one final terrain rendered by point cloud with resolution  $1024 \times 1024$  are shown in Fig. 10, in which subfigures (a) and (b) are observations from different perspectives. From the perspective of Fig. 10(a), we can clearly see the complex mountain features containing ridges and the outspread mountain ranges. As in the perspective of Fig. 10(b), the hierarchy structure of the main streams and tributaries can be clearly recognized. So, the requirements of both mountains and valleys in real scenes can be met by our terrain modeling method. Moreover, the shapes of mountains and valleys in the generated terrain are realistic, because our modeling method conforms to hydrology theory and the skeleton features are enriched by MD and MDI. Compared with the work of Génevaux *et al.* [12] that can also produce large scale terrains based on hydrology, our method generate Tokunaga river network that can make our terrain better conform to the real terrain morphology. Moreover, we use parallel diffusion equation to compute elevations of data points with high resolution in real time. So our method is more suited for generating very large scale terrain with complex features than the work of Génevaux *et al.* The performance of our method is discussed in Section 8.

Through using open software Terragen, we generate various scenes based on our produced terrains. As shown in Fig. 11, the results are realistic and present various geomorphic features. Additionally, the generated terrain can describe a large scale terrain scene, through our procedural modeling method. Obviously, our generated terrain can meet demands in various fields, such as researches in geography, simulation systems and movies.

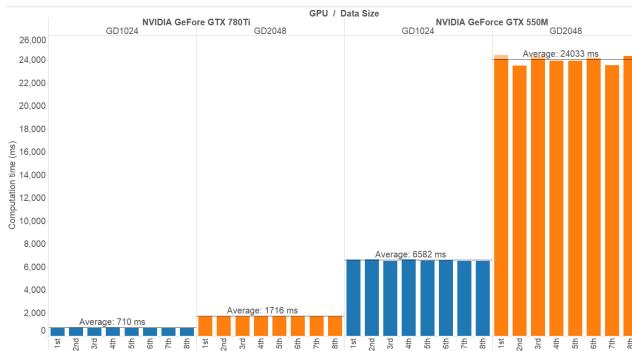


**FIGURE 11.** Terrain scenes generated by software Terragen based on our produced terrains. (a) Terrain scene focused on the ridges. (b) Terrain scene focused on the rivers. (c) Terrain scene from perspective of overlook.

### VIII. PARALLEL IMPLEMENTATION AND PERFORMANCE

Due to the computations of diffusion equation for each grid point are independent, we parallelize the computation of each point in one iteration step on GPU to enhance the performance of terrain diffusion, using OpenCL. We consider the discrete elevation points of terrain as pixels of an image, and store these elevations in the cache of GPU. As the row index and column index of a point are obtained, the computation of Laplace equation is performed in local memory, and the result is written back to the global memory.

We test the performance of diffusion process respectively on Intel core i5-2410M CPU with 2.3Ghz, GPUs of NVIDIA GeForce GT 550M, and NVIDIA GeForce GTX 780Ti, using two terrains with sizes of GD1024( $1024 \times 1024$ ) and GD2048( $2048 \times 2048$ ). We compare the time of executing diffusion equation 5500 times for three hardware environments. The running time on CPU for GD1024 and GD2048 are respectively 323.5s and 21 minutes. The running times on GPUs are shown in Fig. 12. We can see that on NVIDIA GeForce GT 550M using 2 computing units, the average running times are reduced respectively to 6.582s for GD1024 and 24.033s for GD2048. The performance increased nearly 50 times over the performance on CPU. And the running times on NVIDIA GeForce GTX 780Ti using 15 computing units are decreased respectively to 0.71s for GD1024 and 1.726s for GD2048. So our implementation can meet the requirements of real-time terrain generation.



**FIGURE 12.** Comparison of optimization effects on two different GPUs. The computation time of GTX780Ti is much less than GTX 550M. Our method has good scalability.

About other parts in our method, we implement them in CPU without parallel acceleration, because the algorithms are intrinsically efficient. We tested the performance on an Intel core i7-4710MQ CPU with 2.5Ghz. Table 3 lists the

**TABLE 3.** Performance of each part in our method.

Resolution	$N_r$	$N_v$	$T_r$ (ms)	$T_v$ (ms)	$T_d$ (ms)	$T_f$ (ms)	$T_g$ (ms)
GD1024	2033	5968	3.875	820	0	58.6	927.6
GD2048	12267	33291	114.1	32944	13.5	280.1	3642

performance of each part for generating terrains with different sizes.  $N_r$  and  $N_v$  respectively denote the number of river network nodes and Voronoi diagram edges.  $T_r$ ,  $T_v$ ,  $T_d$ ,  $T_f$  and  $T_g$  respectively denote the computation times of river network construction, ridges construction, discretization, feature enrichment and performing Gaussian smoothing 20 times. The timings are the means of performing our method 8 times. The resolution mainly affects the computation time of feature enrichment, and the numbers of river network nodes and Voronoi diagram edges mainly affect the computation for the constructions of river network and ridges. From Fig. 12 and Table 3, we can see that our method can generate large-scale terrain with complex river network efficiently and can meet the requirements of on-line terrain generation.

### IX. CONCLUSIONS AND FUTURE WORK

In this paper, we present a modeling method to generate large scale realistic terrains, by combining hydrology theory with procedural modeling. Our method has the following advantages:

- Large scale terrain can be generated efficiently by using procedural modeling.
- Providing good controllability for the morphology of generated terrains.
- Supporting the augmented visual terrains that contain real terrain features.
- Providing easy-to-use interactive means to set the properties of terrain generation.

In our method, we propose a rule set of L-system to generate Tokunaga river network. Based on the river network, we extract ridges to form more skeleton features that are pivotal to constrain the morphology of the generated terrain. To obtain natural mountains and valleys, we exploit Mid-point Displacement method and its inverse process to enrich the skeleton features. Based on the discrete data points of enriched skeleton features, diffusion equation is utilized to obtain the whole terrain. We develop a technique of generating augmented visual terrains, which allows users to embed real terrain features into the virtual terrain seamlessly through Poisson equation. The performance of our method is enhanced by parallel implementation of terrain diffusion. The terrains generated by our method can present good authenticity, and can describe various terrain scenes, such as mountains, valleys, and waterways.

Since our modeling method only considers the structure information of river network, we want to further quantize the flow rate of the river network in future, for generating more realistic riverways. Based on the methods proposed in this paper, we want to further study the reconstruction of real river scenes.

## REFERENCES

- [1] 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.
- [2] A. Santamaría-IbirikaEmail *et al.*, "Procedural approach to volumetric terrain generation," *Visual Comput.*, vol. 30, no. 9, pp. 997–1007, Sep. 2014.
- [3] V. A. D. Passos and T. Igarashi, "A first person point-of-view example-based terrain modeling approach," in *Proc. Int. Symp. Sketch-Based Int. Modeling (SBIM)*, 2013, pp. 61–68.
- [4] F. P. Tasse, A. Emilien, M. P. Cani, S. Hahmann, and N. Dodgson, "Feature-based terrain editing from complex sketches," *Comput. Graph.*, vol. 45, pp. 101–115, Dec. 2014.
- [5] L. Cruz, L. Velho, E. Galin, A. Peytavie, and E. Guerin, "Patch-based terrain synthesis," in *Proc. 10th Int. Conf. Comput. Graph. Theory Appl. (VISIGRAPP)*, 2015, pp. 189–194.
- [6] O. Št'ava, B. Beneš, M. Brisbin, and J. Křivánek, "Interactive terrain modeling using hydraulic erosion," in *Proc. Siggraph/Eurograph. Symp. Comput. Animation (ACM)*, 2008, pp. 201–210.
- [7] S. D. Peckham, "New results for self-similar trees with applications to river networks," *Water Resour. Res.*, vol. 31, no. 4, pp. 1023–1029, Apr. 1995.
- [8] G. Cui, B. Williams, and G. Kuczera, "A stochastic Tokunaga model for stream networks," *Water Resour. Res.*, vol. 35, no. 10, pp. 3139–3147, Oct. 1999.
- [9] D. Ebert, K. Musgrave, and D. Peachey, *Texturing and Modeling: A Procedural Approach*, vol. 24. San Mateo, CA, USA: Morgan Kaufmann, 1998, pp. 777–786.
- [10] A. Fournier, D. Fussell, and L. Carpenter, "Computer rendering of stochastic models," *Communications*, vol. 25, no. 6, pp. 371–384, Jun. 1982.
- [11] K. Perlin, "An image synthesizer," in *Proc. ACM SIGGRAPH*, 1985, pp. 287–296.
- [12] J. D. Génevaux, E. Galin, E. Guérin, A. Peytavie, and B. Benes, "Terrain generation using procedural models Based on Hydrology," *ACM Trans. Graph.*, vol. 32, no. 4, 2013, Art. no. 143.
- [13] L. Zhang, B. Dai, G. Wang, T. J. Li, and H. Wang, "The quantization of river network morphology based on the Tokunaga network," *Sci. China, Ser. D, Earth Sci.*, vol. 52, no. 11, pp. 1724–1731, 2009.
- [14] P. Claps, M. Fiorentino, and G. Oliveto, "Informational entropy of fractal river networks," *J. Hydrol.*, vol. 187, nos. 1–2, pp. 145–156, Dec. 1996.
- [15] S. T. Teoh, "RiverLand: An efficient procedural modeling system for creating realistic-looking terrains," in *Advances in Visual Computing*, (Lecture Notes in Computer Science), vol. 5875, pp. 468–479, 2009.
- [16] P. Prusinkiewicz and M. Hammel, "A fractal model of mountains with rivers," in *Proc. Graph. Int.*, 1993, pp. 128–137.
- [17] A. D. Kelley, M. C. Malin, and G. M. Nielson, "Terrain simulation using a model of stream erosion," *Acm Siggraph Comput. Graph.*, vol. 22, no. 4, pp. 263–268, Aug. 1988.
- [18] F. Belhadj and P. Audibert, "Modeling landscapes with ridges and rivers: Bottom up approach," in *Proc. 3rd Int. Conf. Comput. Graph. Interact. Techn. Austral. Southeast Asia (Southeast)*, 2005, 447–450.
- [19] F. K. Musgrave, C. E. Kolb, and R. S. Mace, "The synthesis and rendering of eroded fractal terrains," *Comput. Graph.*, vol. 23, no. 3, pp. 41–50, Jul. 1989.
- [20] N. Chiba, K. Muraoka, and K. Fujita, "An erosion model based on velocity fields for the visual simulation of mountain scenery," *J. Vis. Compu. Animation*, vol. 9, no. 4, pp. 185–194, Oct. 1989.
- [21] B. Bedřich and R. Forsbach, "Visual simulation of hydraulic erosion," *J. WSCG*, vol. 10, no. 1, pp. 79–86, 2002.
- [22] B. Beneš, V. Těšínský, and J. Hornýš, "Hydraulic erosion," *Comput. Animation Virtual Worlds*, vol. 17, no. 2, pp. 99–108, Mar. 2006.
- [23] C. Wojtan, M. Carlson, P. J. Mucha, and G. Turk, "Animating Corrosion and Erosion," in *Proc. Eurograph. Conf. Natural Phenomena Eurograph. Assoc.*, 2007, pp. 15–22.
- [24] P. Kříštof, B. Beneš, J. Křivánek, and O. Št'ava, "Hydraulic erosion using smoothed particle hydrodynamics," *Comput. Graph. Forum*, vol. 28, no. 2, pp. 219–228, Apr. 2009.
- [25] X. Mei, P. Decaudin, and B. G. Hu, "Fast hydraulic erosion simulation and visualization on GPU," in *Proc. 15th Pacific Conf. Comput. Graph. Appl.*, 2007, pp. 47–56.
- [26] J. Vanek, B. Beneš, A. Herout, and O. Stava, "Large-scale physics-based terrain editing using adaptive tiles on the GPU," *IEEE Comput. Graph. Appl.*, vol. 31, no. 6, pp. 35–44, Nov./Dec. 2011.
- [27] H. Hnaidi, E. Guérin, S. Akkouche, A. Peytavie, and E. Galin, "Feature based terrain generation using diffusion equation," *Comput. Graph. Forum*, vol. 29, no. 7, pp. 2179–2186, Sep. 2010.
- [28] V. Skytt, O. Barrowclough, and T. Dokken, "Locally refined spline surfaces for representation of terrain data," *Comput. Graph.*, vol. 49, pp. 58–68, Jun. 2015.
- [29] K. A. Johannessen, T. Kvamsdal, and T. Dokken, "Isogeometric analysis using LR B-splines," *Comput. Methods Appl. Mech. Eng.*, vol. 269, pp. 471–514, Feb. 2014.
- [30] B. Rusnell, D. Mould, and M. Eramian, "Feature-rich distance-based terrain synthesis," *Visual Comput.*, vol. 25, nos. 5, pp. 573–579, May 2009.
- [31] H. Zhou, J. Sun, G. Turk, and J. M. Rehg, "Terrain synthesis from digital elevation models," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 4, pp. 834–848, Jul./Aug. 2007.
- [32] J. Gain, P. Marais, and W. Straßer, "Terrain sketching," in *Proc. ACM SIGGRAPH Symp. Interact. 3D Graph. Games*, 2009, pp. 31–38.
- [33] R. E. Horton, "Erosional development of streams and their drainage basins, hydrophysical approach to quantitative morphology," *Bull. Geological Soc. Amer.*, vol. 56, no. 3, pp. 275–370, 1945.
- [34] M. I. Shamos and D. Hoey, "Geometric intersection problems," in *Proc. 17th Annu. Symp. Found. Comp. Sci.*, 1976, pp. 208–215.
- [35] Y. Yu *et al.*, "Mesh editing with poisson-based gradient field manipulation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 644–651, 2004.



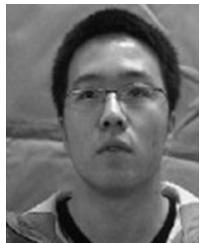
HUIJIE ZHANG (M'16) received the Ph.D. degree from Jilin University in 2009. She is currently an Associate Professor with the School of Computer Science and Information Technology, Northeast Normal University. Her main research interests include scientific visualization, information visualization, visual analysis, computer graphics, 3D model simplification, multi-resolution modeling for terrain and 3-DGIS and optimization algorithm. She is a member of Computer Federation.



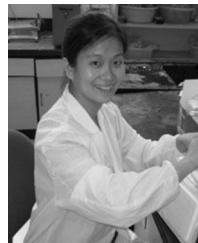
DEZHAN QU received the B.Sc. degree from Northeast Normal University in 2014. He is currently pursuing the M.Sc. degree with the School of Computer Science and Information Technology, Northeast Normal University. His current research interests include computer graphics and scientific visualization.



YAFANG HOU received the B.Sc. degree from Northeast Normal University in 2014. She is currently pursuing the M.Sc. degree with the School of Computer Science and Information Technology, Northeast Normal University. Her current research interests include computer graphics and scientific visualization.



**FUJIAN GAO** received the B.S. degree from Heilongjiang University in 2013 and the M.Sc. degree from the School of Computer Science and Information Technology, Northeast Normal University. His current research interests include computer graphics and scientific visualization.



**FANG HUANG** received the Ph.D. degree from the Northeast Institute of Geography and Agroecology, Chinese Academy of Sciences, in 2001. She is currently an Associate Professor with the School of Geographical Science, Northeast Normal University. Her main research interests include geographical information system, resource and environment remote sensing, and scientific visualization.

• • •