

基于概要图的数字地形生成方法^①



贾浩霖, 王妍, 王建一

(哈尔滨工业大学 建筑学院 互动媒体设计与装备服务创新文化和旅游部重点实验室, 哈尔滨 150001)

通信作者: 王建一, E-mail: wangjy@hit.edu.cn

摘要: 针对地形生成算法在易用性、可控性、真实感、速度等方面难以平衡的问题, 提出了一种基于概要图的地形生成方法. 该方法使用生成对抗网络, 在隐空间中通过对地形坡度、坡向等信息建模, 使生成的地形符合用户手绘概要图约束. 提出基于地形高度图的概要图提取算法, 可从真实地形高度图中提取出类似用户手绘效果的概要图, 快速构建数据集. 提出多尺度地形细节填充算法, 通过引入多尺度的概念, 使地形纹理细节得到了动态补充, 真实感美观度大幅提升. 提出了基于用户反馈的地形满意度评价方法, 并据此进行实验验证, 结果表明该地形生成方法可以准确高效地生成符合用户预期的数字地形.

关键词: 概要图; 地形生成; 细节填充; 生成对抗网络

引用格式: 贾浩霖, 王妍, 王建一. 基于概要图的数字地形生成方法. 计算机系统应用, 2023, 32(1): 327-336. <http://www.c-s-a.org.cn/1003-3254/8878.html>

Digital Terrain Generation Method Based on Sketch Map

JIA Hao-Lin, WANG Yan, WANG Jian-Yi

(Key Laboratory of Interactive Media Design and Equipment Service Innovation of Ministry of Culture and Tourism, School of Architecture, Harbin Institute of Technology, Harbin 150001, China)

Abstract: As the terrain generation algorithm has trouble balancing ease of use, controllability, realism, and speed, this study proposes a terrain generation method based on sketch maps. This method uses the generative adversarial network to model the terrain slope, slope aspect, and other information in the hidden space so that the generated terrain conforms to the constraints of the user's hand-drawn sketch map. This study also proposes a sketch map extraction algorithm based on terrain height maps, and it can extract a sketch map to a hand-drawn effect from a real terrain height map and quickly build the data set. An algorithm for multi-scale terrain detail filling is proposed. Owing to the introduction of the multi-scale concept, the terrain texture details are dynamically supplemented, and the realism and aesthetic properties are greatly improved. A terrain satisfaction evaluation method based on user feedback is proposed and verified by experiments. The results show that the proposed terrain generation method can accurately and efficiently generate digital terrain that meets users' expectations.

Key words: sketch map; terrain generation; detail filling; generative adversarial network (GAN)

地形生成一直以来都是计算机图形学中的重要研究对象, 其成果广泛应用于仿真、军事、电子游戏、影视等领域. 如何通过直观简洁的交互, 帮助设计师快速生成高质量且形态可控的地形成为一个重要课题. 该技术能够有效地帮助设计师从繁琐的地形细节制作

中解脱出来, 更多地着眼于地形起伏、走势等关键环节的设计.

地形生成任务主要涉及 3 个核心问题, 分别是地形细节生成、地形形态控制以及地形存储表示, 其中地形细节生成和形态控制方法是本文主要研究的问题.

^① 收稿时间: 2022-04-28; 修改时间: 2022-06-01; 采用时间: 2022-06-13; csa 在线出版时间: 2022-08-19

CNKI 网络首发时间: 2022-11-15

细节生成主要探讨如何让生成的地形更逼真,而形态控制则探讨如何让生成地形的走势、布局、地貌风格等宏观信息符合设计师给定的约束条件。

现有的地形生成方法往往难以兼顾细节生成和宏观控制,造成地形编辑器学习门槛高、使用效率低等问题。针对这一现象,本文提出了基于概要图的地形生成算法,该方法首先构建了大批量的数据集,并结合生成对抗网络,隐式地习得概要图到地形高度图的映射关系,实现了由概要图到真实地形的生成过程。为了进一步提高地形真实感,本文还提出了多尺度的地形细节填充算法,根据地形中不同区域不同的纹理稀疏程度,动态地进行细节填充。最后本文使用 Python 自带的 Tkinter 库为本算法实现了地形编辑器窗口程序,并提出了基于用户反馈的地形满意度指标,在实验中验证了本算法的有效性。

1 相关工作

目前,地形细节生成方法大致可以分为3类:基于过程式的、基于仿真模拟的和基于实例的^[1],地形形态控制方法大致可以分为两类:基于参数的,以及基于概要图或曲线的。

基于过程式的生成方法通常借助噪声函数或细分网格算法,可以快速生成细节良好且风格各异的大型地形^[2,3],但该类方法一般随机性较强,通常使用基于参数的控制方法对地形整体风格进行控制,比如将地形生成算法中的参数暴露给设计师,或者通过定义扭曲函数对随机生成的地形进行扭曲^[4],但设计师难以对地形的起伏及走势进行控制。部分方案使用多项式曲线的控制方法对过程式的生成效果进行约束^[5],但由于曲线随机感较差,生成效果不够真实。

基于仿真模拟的生成方法可以创造出符合地质学特征的地形,细节逼真^[6-8],但该类方法计算成本较高,且可控性较差,往往用于优化细节。

基于实例的生成方法大多通过提取和拼接真实世界的扫描数据来生成地形,并可通过概要图对地形走势进行约束,能够较好地满足可控性和真实感^[9],但这类算法不能产生实例以外的新特征,且容易产生重复纹理。

近年来深度学习技术在图像跨域转换问题上取得了令人瞩目的成绩。Isola 等人^[10]基于 cGAN 提出的 pix2pix 网络,有效解决了有监督的图像跨域转换问题,可以对基于线稿或色块的概要图进行细节丰富,生成

符合概要图约束的逼真图像,为带有条件约束的生成问题提供了可行的解决方案。Guérin 等人^[11]将 pix2pix 网络应用于地形生成领域,对基于概要图的地形生成任务进行了初步探索。

2 算法总体框架

2.1 问题描述

本算法的输入是一张用户绘制的概要图,概要图上可以使用线条表示某个地形走势,并使用不同颜色代表不同的地形类型,也可以使用其他形状图案表示额外的约束信息,以更精准的描述地形要求。本文将设计一个算法,根据概要图生成一个符合其约束的真实感地形,并将该地形的高度图作为算法输出。生成的地形应满足以下两个条件:1) 严格满足概要图对不同地形类型及其走势或起伏的约束;2) 适当地添加额外的地形细节,以获得完整且富有真实感的地形。

2.2 算法思路及流程

三维地形可以被表示为一张二维高度图,而用户绘制的概要图也是一张二维图片,因此该问题也可以被抽象为图像到图像的映射问题。并且由于地形分布具有较强的统计学规律,因此可以应用神经网络模型对概要图到高度图的映射关系隐式地进行建模, pix2pix 模型为解决这类图像跨域转化问题提供了很好的解决方案^[10]。

除了模型,数据集的构建也是影响生成效果的关键因素。真实地形的高度图可以从卫星扫描数据中获得,而高度图对应的概要图则需要使用算法进行生成,因为很难人工地绘制上万张概要图。针对概要图生成问题,本文提出使用河流仿真等算法进行多种地形的概要图生成,并通过一定的改造,使生成的概要图更加接近用户手绘效果,提高模型训练效果。除此之外,在数据集的制作问题上,本文通过引入多稀疏尺度的概要图生成手段,使模型对不同稀疏程度的概要图输入都有很好的生成效果。

另一方面,直接由神经网络生成的地形虽然整体效果很好,但往往细节缺失。为了进一步优化地形细节,提高地形表现力,本文提出了多尺度地形细节填充算法。其核心思路是根据地形复杂程度,将低分辨率地形高度图进行多尺度分割,然后为每个分割块寻找最相似的重建方案,并替换为该方案对应的高分辨率地形块,以达到提升地形细节效果的目的。通过引入多尺度

的算法思想,该算法解决了单一尺度下山脉与平原纹理风格趋于相似的问题,实现了更自然多样的重建效果。

整体算法流程如图1所示。

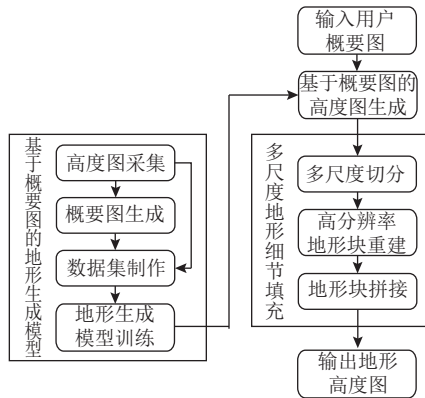


图1 整体算法流程图

3 基于概要图的地形生成模型

3.1 数据集构建

3.1.1 真实地形高度图的获取与处理

为了保证模型最后的生成数据符合真实地形分布规律,需要获取大量真实地形数据作为训练集.本文中使用的真实地形数据来自于 SRTM 网站,精度为 90 m,数据格式为 tif 格式,位深度为 16 位,数据是中国境内的地形数据.将原始的高度图以 128 像素为步长,切分成 256×256 分辨率的高度图块,得到用于训练的地形高度图数据.值得注意的是,数据集中若包含大量风格差异较大的地貌类型,则会引起训练过程不稳定,模型误差难以收敛,训练得到的模型输出结果较模糊等问题,因此需要一张等比例的地形类型图作为是否提取某一地形块的判断依据,只有当该地形块大部分区域为丘陵或者山地时,该地形块才会被提取并作为数据集制作的原材料.经过上述操作,本文共提取了 1 万余个地形块用于后续数据集制作。

为了保证训练的稳定性,需要统一所有高度图的平均灰度.计算出每张图像所有像素的均值 M ,然后对每个像素值 x 按照式(1)得到处理后的值 y .效果如图2所示。

$$y = x + (128 - M) \quad (1)$$

3.1.2 概要图提取算法需求分析

神经网络模型的训练是以数据为导向的,数据集很大程度上决定了模型的功能,因此需要对概要图中需要包含的信息进行分析,以便后续概要图提取算法

的设计。

对于任意一种地形,我们可以把影响地形样貌的因素分解为高程、坡度、坡向这3个变量.对于高程,算法只需要生成相对 0 海拔的高程即可,因此坡度和坡向是影响地形生成样貌的关键指标,比如蜿蜒的山脉坡向变化剧烈,而平原相对变化幅度较小。

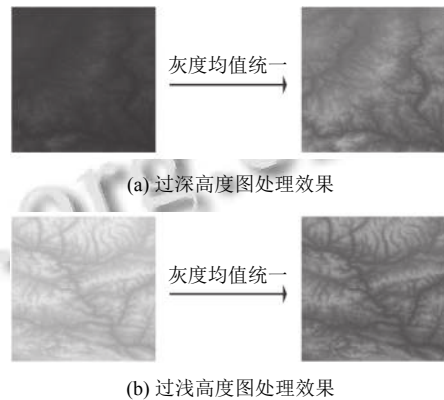


图2 高度图均值统一后效果

坡向往往与地形走势近似垂直,因此绘制用于描述地形走势的曲线是必要的.对于坡度的控制,可以有多种方法,其中最简单的是线性缩放.但简单线性缩放难以对不同区域进行差异化控制.在概要图中加入海拔信息是一个可行的方案,这在 Guérin 等人^[1]提出交互式地形生成方法中得到了验证。

总的来说,概要图中应该包括描述山川河流走势的曲线,以及海拔控制点,这些信息将帮助算法的使用者更好地控制地形的宏观形态.为此本文将概要图拆分为如下3个组成部分:河流网特征图,山脊线特征图,以及海拔控制点特征图。

3.1.3 河流网特征图提取

河流网图的生成方法主要依靠基于高度图的河流仿真算法实现,比较常用的方法是 D8 算法^[12].该算法利用最大坡降法来计算高度图中每个像素点与其领域的 8 个像素点的距离权落差,将距离权落差最大的方向作为该栅格的流向.本文中的河流网提取算法参考传统 D8 方法进行水流模拟.通过设定一定的条件选取河流系统的起始像素,分析这些像素周围的坡度来找到水的流动方向,反复迭代,实现对水流汇聚过程的仿真模拟.本文选择从高度图边缘创建河流系统,以获得较为干净准确的河流,这有利于提取出更贴近用户手绘效果的河流网特征图。

对于平原等起伏不明显的地貌,直接使用河流仿

真算法难以提取出准确的河流图像,因此需要对所有高度图进行线性灰度扩展来增加对比度,将灰度范围扩展为0至255.首先计算出高度图所有像素中的最大值 max 及最小值 min ,然后使用式(2)对每个像素值进行更新.其中, y 为灰度扩展后该点的像素值, x 为灰度扩展前该点像素值.

$$y = ((x - min) \div (max - min)) \times 255 \quad (2)$$

河流网的提取需要满足一定的阈值约束,某一像素需要汇聚足够多的“水”才能被认为是河流的组成部分,选取合适的阈值可以过滤掉多余的细小支流.本文中,每个像素点积水量的范围为0到255.为了保证不同稀疏度的河流网输入都能生成较为完整的地形,本文选取128、160和192三个阈值生成不同稀疏尺度

的河流网.实验证明,多尺度的训练数据可以显著增强模型对不同稀疏度概要图的适应能力.

由于用户实际绘制的河流图像往往是平滑且富有概括性,为了避免过拟合,需要在河流仿真前对高度图进行降采样处理和模糊处理,以提取出更具有概括性的河流网图.本文首先对 256×256 分辨率的高度图进行一次降采样生成 128×128 分辨率图像,然后使用 2×2 窗口进行高斯模糊,再进行一次降采样生成 64×64 分辨率图像,最后进行一次 5×5 窗口的高斯模糊.对模糊降采样之后的图像进行河流仿真,得到 64×64 分辨率的河流网图,然后进行两次升采样得到 256×256 分辨率河流网图,最后使用骨骼算法提取出宽为1个像素的河流骨骼图像.具体流程如图3所示.

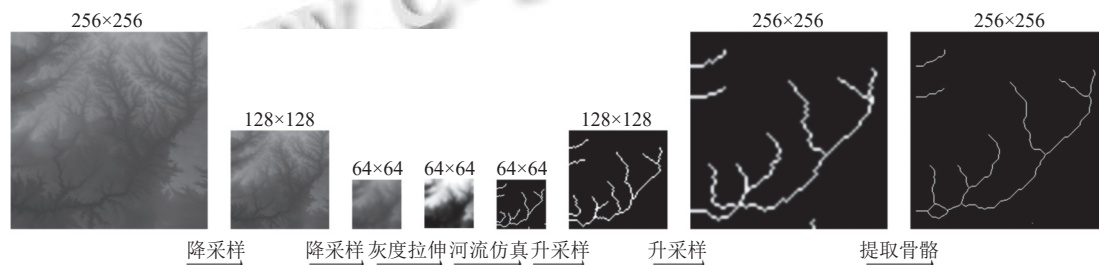


图3 河流网图提取流程

3.1.4 山脊线特征图提取

与河流网图的生成方法类似,但需要在执行河流仿真算法前将高度图的灰度值取反相,这样生成的河流网图像与原图的山脊线图像是一致的.效果如图4所示.

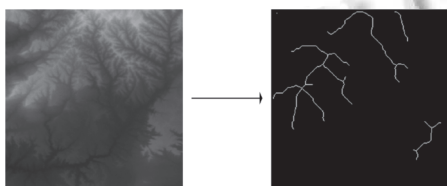


图4 山脊线图提取效果

3.1.5 海拔控制点特征图提取

本文在海拔控制点数据集构造过程中,设置了低海拔和高海拔两种海拔控制点,分别用50的灰度值和204的灰度值的 3×3 小矩形块代表.

本文通过如下几条规则找出某一灰度均值统一后的高度图中的海拔控制点,在这里只对高海拔控制点

的提取方法做描述,低海拔控制点提取方法与之类似.

- 1) 待选的高海拔控制点的像素值要高于90%的本图像素值.
- 2) 待选的高海拔控制点的灰度值不能小于148.
- 3) 待选的高海拔控制点周围至少有4个点比其海拔低.
- 4) 最终的高海拔控制点要在待选高海拔控制点中随机选择.
- 5) 待选的高海拔控制点灰度值越大,被选择的概率越高.
- 6) 待选的高海拔控制点越少,每个控制点被选择的概率越大.
- 7) 每个控制点之间的距离必须大于设定的阈值.

按照如上原则,最终生成的海拔控制点图如图5.

3.1.6 特征图组合

到此为止本文已经阐述了河流网图、山脊线图以及海拔控制点图的构造方法,接下来将介绍如何将

度图拼接构成数据集.

首先我们使用一张三通道的 RGB 彩色图片存储概要图, 河流网特征图放入 RGB 图像的 B 通道中, 山脊线特征图放入 R 通道中, 海拔控制点特征图放入 G 通道中. 另一方面, 用户在绘制概要图时, 不一定会使用全部的特征图作为控制手段, 因此为防止过拟合, 需要随机删去 RGB 图片 3 个通道中的某几个通道中的数据, 并至少保留 1 个通道中存有数据.

按照如上操作, 本文最终构建了 1 万余个可用的数据集, 部分数据集数据如图 6 所示.

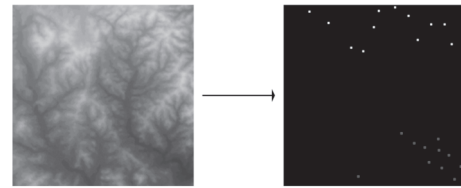


图 5 海拔控制点图提取效果

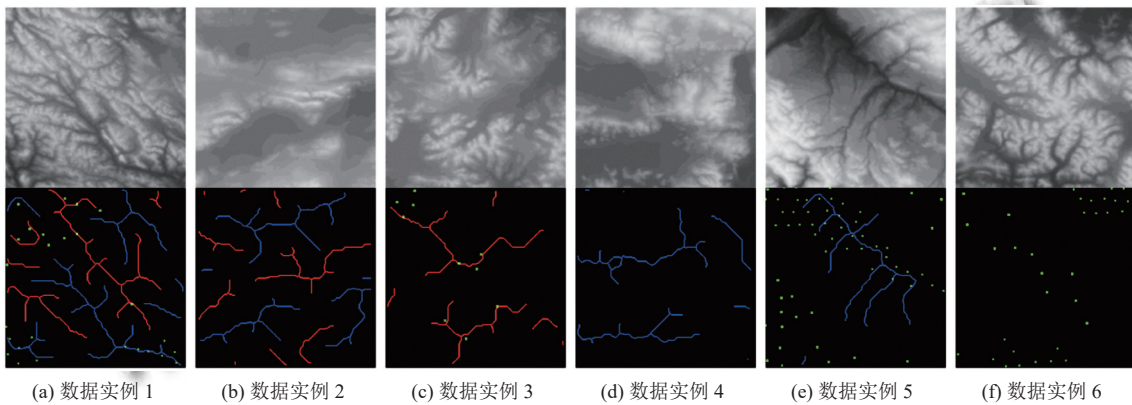


图 6 部分数据集效果

3.2 地形生成神经网络

本文采用的神经网络模型主要基于 pix2pix 模型, 隐式地对概要图到地形高度图之间的映射关系进行建模, 当用户输入自己的特征概要图时, 模型生成符合其约束的真实感地形高度图.

pix2pix 模型主要包括两个网络, 生成器 G 和判别器 D , 生成器 G 将输入的条件图映射成目标图像, 判别器 D 判断图片真伪及是否符合条件图的约束. 二者在损失函数和训练次序的共同作用下, 互相博弈, 实现生成器参数的最优化^[10]. 整体网络结构如图 7 所示.

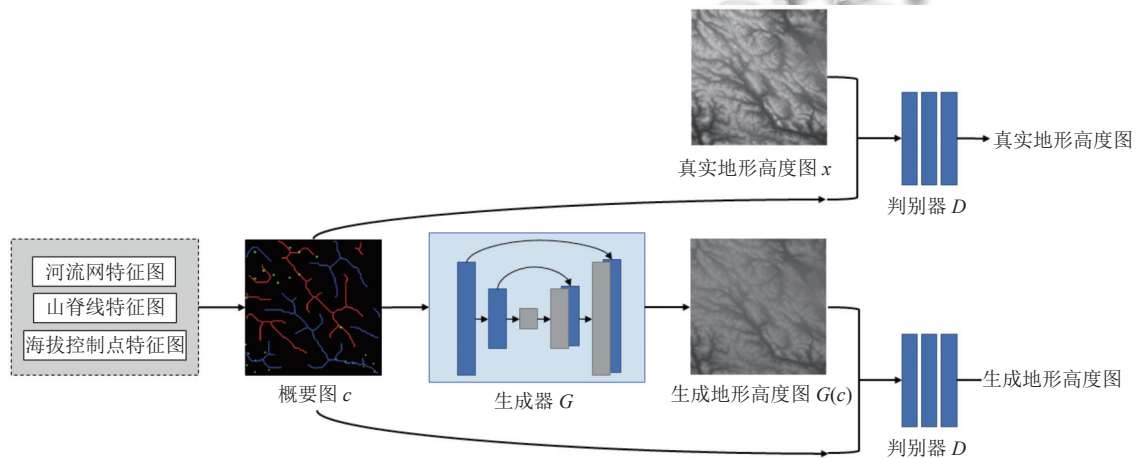


图 7 地形生成任务整体网络架构

具体而言, 生成器由编码器和解码器组成, 这样的编解码结构有助于网络在隐空间中对坡度、坡向信息建模. 编码器的输入是一个 $256 \times 256 \times 3$ 的图像 c , 第

1 层使用 64 个步长为 2 的 $4 \times 4 \times 3$ 卷积核, 并用 a 等于 0.2 的 LReLU 函数激活; 第 2 层到第 4 层使用 $4 \times 4 \times 3$, 步长为 2 的卷积核, 且卷积核数量逐层翻倍, 在每次卷

积完成后进行批归一化,并用 a 等于 0.2 的 LReLU 函数激活;第 5 层到第 8 层卷积核数量不再翻倍,其余保持不变,最终得到了一个 $1 \times 1 \times 512$ 维的特征图。解码部分执行和编码器基本对称的反卷积操作,区别是此时的激活函数为 ReLU,最后一层使用 tanh 函数激活,以保证训练稳定性,并在前 3 层使用 Dropout 避免过拟合。此外,借鉴了 U-Net 网络结构,对称的编码层和解码层之间使用残差连接,帮助解码器恢复细节信息^[13]。最后生成器的输出是一个 $256 \times 256 \times 3$ 的生成图像 $G(c)$ 。

判别器需要将条件图 c 分别与真实地形高度图 x 和生成器所生成的假图 $G(c)$ 拼接,形成 $256 \times 256 \times 6$ 的图像,然后传入判别器判断真假。如果 c 是与 x 拼接的,那么判别器结果应趋向为 1,若与 $G(c)$ 拼接,则应趋向为 0。判别器拥有多个卷积层,结构与生成器的编码器类似,最后一层的激活函数使用 Sigmoid 函数,最后生成 $30 \times 30 \times 1$ 的张量,其各个项的均值表示判别器的判别结果。

损失函数是 L1 损失和 cGAN 损失相结合。pix2pix 原作者在实验中发现,仅使用 L1 损失可以生成可靠的低频信息,但高频缺失,图像模糊;仅使用 cGAN 的损失函数虽然可以生成较为清晰的图像,但是会产生大量伪影。pix2pix 作者将两者相结合,得到了可以生成较为理想图像的损失函数^[10]。设损失函数 G^* ,其具体定义如式(3)–式(5)所示:

$$G^* = \arg \min_G \max_D L_{cGAN} + \lambda L_{L1} \quad (3)$$

$$L_{cGAN} = E_{c,x} [\log D(c,x)] + E_c [\log(1 - D(c,G(c)))] \quad (4)$$

$$L_{L1} = E_{c,x} [\|x - G(c)\|_1] \quad (5)$$

在训练过程中,对判别器 D 的训练需要最大化 L_{cGAN} 以尽可能区分真伪图,并判别是否条件图约束;对生成器 G 训练需要最小化 L_{cGAN} ,以生成更加优质的图像。

4 多尺度地形细节填充算法

4.1 地形细节填充算法描述

算法的输入是由神经网络模型直接生成的细节较少的高度图,输出是整体形态一致,细节丰富的细节填充后的高度图。

地形细节填充算法可以划分为 3 个部分:切分、重建和拼接。切分任务的主要目的是将原始高度图以某种方式切分成若干个用于重建的小块,切分地越细密,重建后的地形与原始地形越相似,同样也意味着越

难填充进足够的细节。重建任务是使用最相似的高分辨率地形块替换掉原始的低分辨率地形块,实现对细节的丰富。拼接任务则是将重建后的地形块,按照原位置无缝地融合在一起,来得到细节填充后的完整地形。

4.2 单一地形块重建

多尺度细节填充的基础是对单一地形块进行重建。在本节中将主要介绍如何构建双分辨率字典,并使用该字典实现单一地形块重建。

4.2.1 双分辨率地形字典构建

地形字典是由一维原子向量排列成的二维矩阵,每个原子都是一个由真实地形块变形成的一维向量,真实地形块从遥感高度图中采样得到。

由于算法目的是实现分辨率的提升和细节的丰富,因此需要建立孪生的高分辨率和低分辨率两个地形字典,低分辨率字典的原子是对应高分辨率字典的原子缩小并平滑后的结果。低分辨率字典中的原子应与待重建地形块变形成一维向量后的尺寸相同,这样可以直接使用向量内积计算所有原子与待重建地形块的相似度,找到最优原子;而高分辨率字典用于实现细节填充。

为了优化算法的稳定性,每对高低分辨率原子都应除以低分辨率原子的模长,以实现低分辨率原子的归一化,并保证重建过程中求得的系数对高分辨率原子同样有意义。

4.2.2 地形块重建

地形块重建的核心任务是从低分辨率字典中找到用于重建目标地形块的最优原子,并替换为对应的高分辨率地形块,因此其核心算法如算法 1。

算法 1. 单一地形块重建算法

- 1) 输入: 大小为 $N \times n$ 的待重建地形块矩阵 Y , 大小为 $N \times n^2$ 的低分辨率地形字典矩阵 D_{low} , 大小为 $N \times m^2$ 的高分辨率字典矩阵 D_{high} , 其中 D_{low} 和 D_{high} 的每一个行向量代表一个原子。
- 2) 将地形块矩阵 Y 变形成大小为 $n^2 \times 1$ 的列向量 y 。
- 3) 找到列向量 y 与 D_{low} 的所有行向量中,内积绝对值最大项的索引 λ , D_{low} 的第 j 个行向量用 $d_{low}(j)$ 表示。
- 4) 获取矩阵 D_{low} 的第 λ 行向量 $d_{low}(\lambda)$, 取其逆得列向量 $d_{low}(\lambda)^{-1}$ 。记重构系数为 a , 则 a 等于 y 与 $d_{low}(\lambda)^{-1}$ 的内积。
- 5) 获取 D_{high} 的第 λ 行高分辨率原子 $d_{high}(\lambda)$, 将其乘以系数 a , 最后变形成大小为 $m \times m$ 的细节填充后地形块 Y^* 。
- 6) 输出细节填充后地形块 Y^* 。

作为实验,使用该算法将 16×16 的地形块重建为 128×128 大小。重建前后效果对比如图 8 所示,可以看出重建后的地形块整体上与之前的地形块是相似的,且细节纹理更加丰富。

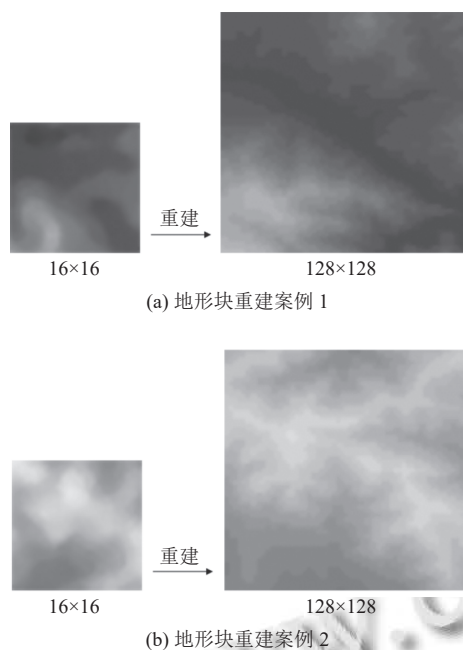


图8 地形块重建前后效果对比

4.3 多尺度重建算法

多尺度重建的思想是: 将 256×256 的待重建地形块分为 64×64 、 16×16 、 8×8 三个级别, 划分依据为地形块的均方差, 也就是该地形块的细节复杂程度. 对于均方差过高的大尺度地形块, 需要将其细分为中尺度地形块以降低每个地形块的复杂度, 中尺度到小尺度的细分过程同理. 由于不同尺度的地形块大小不同, 重建需要的纹理类型也不尽相同, 因此需要准备 3 个尺度下的双分辨率地形字典. 算法核心步骤如算法 2.

算法 2. 多尺度重建算法

- 1) 输入: 待切分地形高度图 T . 大中小 3 个尺度的双分辨率字典 D_{low}^h 、 D_{high}^h 、 D_{low}^m 、 D_{high}^m 、 D_{low}^l 、 D_{high}^l . 大中小 3 个尺度下地形块边长 s_h 、 s_m 、 s_l . 用于划分大尺度与中尺度边界的均方差值 σ_{hm} , 划分中尺度与小尺度边界的均方差值 σ_{ml} .
- 2) 将待切分地形高度图 T 切分成若干个边长为 s_h 的大尺度地形块, 将所有大尺度地形块数据及其在 T 中的位置存储到序列 l_h 中.
- 3) 计算 l_h 序列中所有地形块的均方差是否小于 σ_{hm} , 如果大于 σ_{hm} , 则将其切分成若干个边长为 s_m 的中尺度地形块, 并从 l_h 中删除. 将所有中尺度地形块数据及其在 T 中的位置存储到序列 l_m 中.
- 4) 计算 l_m 序列中所有地形块的均方差是否小于 σ_{ml} , 如果大于 σ_{ml} , 则将其切分成若干个边长为 s_l 的小尺度地形块, 并从 l_m 中删除. 将所有小尺度地形块数据及其在 T 中的位置存储到序列 l_l 中.
- 5) 根据单一地形块重建算法, 使用大中小 3 个尺度的双分辨率字典, 分别对序列 l_h 、 l_m 、 l_l 中的地形块重建, 并将序列中存储的地形块位置按地形块放大比例重映射.
- 6) 输出序列 l_h 、 l_m 、 l_l .

利用该算法进行多尺度切分, 其切分效果如图 9

所示. 从图 10 中可以看出, 使用该算法进行重建, 地貌纹理的丰富度会大大提高, 并且相比于单尺度重建, 整体上与重构前的地形更加吻合, 小尺度的细节也被很好地重建出来.

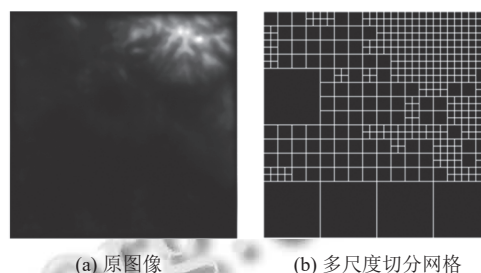


图9 多尺度切分效果

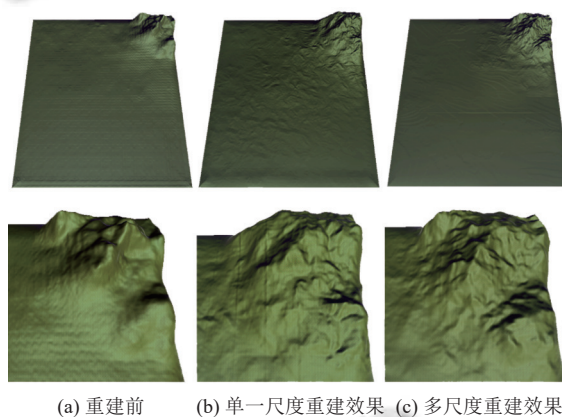


图10 单一尺度与多尺度重建效果对比

4.4 地形块拼接

地形块拼接的目标是让块与块之间毫无违和感地拼接在一起. 本算法参考了 Gènevaux 等人^[14] 提出的多地形融合算法, 该算法扩大了待拼接地形块的边界范围, 在待拼接地形的外围定义了一个环形缓冲带作为与其他地形块拼接时的融合区域. 此外算法还提供了一个权函数, 用来定义地形块不同部位对其他地形块的影响权重. 为了适应需求, 本文在该融合算法的基础上进行了改进, 将整个地形块都看作缓冲带, 并重新定义了权函数. 为了更准确地描述该算法, 需要定义一些符号.

用 s_h 、 s_m 、 s_l 分别表示大中小 3 个尺度地形块的边长大小. 用 p_h 、 p_m 、 p_l 分别表示大中小 3 个尺度地形块扩大边界后的边长大小.

设集合 Ω 是 R^2 的子集, 且 Ω 为紧集, 集合 Ω 可以看作是承载所有待拼接地形块的“画布”. 地形块被定

义为一个定义域在 Ω 上的高程函数 f , 函数 f 的值域为 R . 用 w 表示权函数, w 的定义域为 Ω , 值域为 R . 因此, 如果设有 N 个待拼接地形块, 且第 i 个待拼接地形块用 f_i 表示, 第 i 个待拼接地形块的权函数用 w_i 表示, 则拼接后的地形 f 可以用式 (6) 表示, 其中, v 是 Ω 中的元素:

$$f(v) = \frac{\sum_{i=1}^N f_i(v) w_i(v)}{\sum_{i=1}^N w_i(v)}, v \in \Omega \quad (6)$$

为了保证权函数的平滑性, 权函数 w 需要有一阶连续导数. 设 c_i 为第 i 个待拼接地形块的中心点, 则 c_i 的两个坐标轴分量可以表示为 $c_i(0)$ 和 $c_i(1)$; 同理, 点 v 的两个坐标轴分量可以用 $v(0)$ 和 $v(1)$ 表示; p_j 为待拼接地形块的尺寸, 则权函数 w_i 可以用式 (7) 表示:

$$w_i = \left(1 - \left(r(v, c_i) \div (p_j \div 2)\right)^2\right)^3, p_j \in \{p_h, p_m, p_l\} \quad (7)$$

$$r(v, c_i) = \max(|c_i(0) - v(0)|, |c_i(1) - v(1)|) \quad (8)$$

通过实际对比可以看到, 本文提出的地形块拼接算法与无缓冲带的简单地形块拼接方法相比, 地形块之间的边界更加自然和谐, 如图 11 所示.

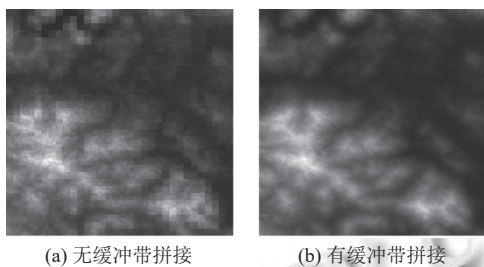


图 11 有无缓冲带的地形拼接效果对比

5 生成效果验证

对地形生成质量进行客观评价是验证生成算法能力的重要组成部分. 由于该算法的目的是辅助设计师进行设计, 所以算法生成效果应由设计师进行评价, 为此本文提出了基于用户反馈的地形满意度指标, 实现对生成效果进行量化评价. 除了地形满意度, 效率也是算法关心的问题, 因此制作地形所需要的时间也是重要的评价指标. 一般来讲, 制作一个地形的满意度越高, 同时所需时间越短, 这个地形生成算法就越好.

5.1 基于用户反馈的地形满意度指标

对于一个由本文算法生成的 $n \times n$ 像素的地形高度图 A , 数据类型为无符号整型且位深度为 d . 设计师在该地形的基础上使用建模工具进行二次编辑, 得到心目中的理想地形, 并导出为 $n \times n$ 像素的地形高度图 B , 数据类型为无符号整型且位深度为 d . 高度图 A 和 B 的第 i 行第 j 列像素值分别用 $a(i, j)$ 和 $b(i, j)$ 表示, 则地形满意度 TS (terrain satisfaction) 可定义为:

$$TS = 1 - \frac{\sum_{i=1}^n \sum_{j=1}^n |a_{ij} - b_{ij}|}{n \times n \times (2^d - 1)} \quad (9)$$

容易看出, 式 (9) 是在 L1 距离的基础上改进而来, $0 \leq TS \leq 1$, 且 TS 越大代表地形满意度越高, 算法生成地形与理想地形之间的差距越小. 为了更客观地反映算法效果满意度, 需要对多个测试样本取均值, 得到地形平均满意度 TAS (terrain average satisfaction), 对于 m 个测试样本, 分别用 TS_1, TS_2, \dots, TS_m 表示, 则 TAS 可定义为:

$$TAS = \frac{\sum_{i=1}^m TS_i}{m} \quad (10)$$

与 TS 指标类似, $0 \leq TAS \leq 1$, 且 TAS 越大代表地形平均满意度越高.

5.2 实验环境

环境配置如下: CPU: AMD R7 2700X; 内存: 16 GB; GPU: NVIDIA RTX 3070; 硬盘: 1 TB SSD; 操作系统: Windows 10; 深度学习平台: TensorFlow 1.14.0. 模型经历 300 个训练周期的训练, 训练时长为 46 h 02 min.

软件方面, 为验证本文算法的可用性, 本文使用 Python 自带的 Tkinter 库, 为算法实现了一个地形编辑器窗口程序, 以便于设计师编辑地形, 地形编辑器的效果如图 12 所示. 对比实验使用 Houdini 19.0 进行基于过程式的地形编辑. 二次编辑得到理想地形的过程是使用 Unity3D 游戏引擎自带的地形编辑工具, 通过笔刷雕刻等方法, 实现对地形外观的高度自定义.

5.3 实验结果与分析

本实验请到了 10 名志愿者进行对比实验, 来验证算法有效性. 第 1 组实验要求志愿者使用基于本文算法实现的地形编辑器制作 40 个地形, 并记录制作时间. 第 2 组对比实验要求志愿者针对给定的地形分布要求,

使用主流的基于过程式的生成方法制作 40 个地形,并记录制作时间.第 2 组实验给定的生成算法包括 Perlin 噪声、Worley 噪声、曲面细分等主流地形生成方法,以及 Lattice 扭曲函数和多种基于噪声的扭曲函数.这些方法均使用 Houdini 软件中提供的算法模块完成.两组实验得到的地形使用位深度为 16 位的 png 格式保存,大小为 2049×2049 像素.最后按照 *TS* 指标要求,在 Unity3D 引擎中修改为理想地形,并计算出两组实验的 *TAS* 值.得到的对比试验结果见表 1,部分实验生成效果见图 13.从 *TAS* 值来看,本文算法的 *TAS* 值略大于过程式生成方法,说明算法所生成的地形需要更少的改动即可满足用户期望.这是由于本文算法通过直观的基于概要图的交互方式,可以使地形生成过程具有更强的约束性,能够更精确地得到期望的地形分布.相比之下,基于过程式的生成方法随机性过强,往往会得到一些不需要的地形区域.另一方面,从花费时间上来看,本文算法 53.2 s 的平均编辑时间意味着对于大多数地形,设计师可以快速得到想要的效果,这相较于使用 Houdini 进行的基于过程式的地形编辑所需要的 622.2 s 而言有了巨大提升,这是由于基于概要图的地形编辑方式及人工智能辅助设计手段,大大降低了软件使用门槛,提升了地形编辑效率.

为了进一步直观感受本文所提算法在实际应用中的效果,在图 14 中,本文在 Unity3D 引擎中,使用由该

算法直接生成的地形,搭建了一个森林场景,进一步验证了该算法能够保证美观性和真实感的要求.

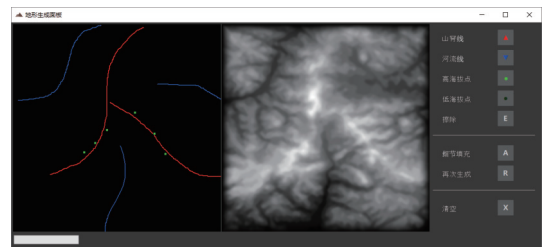
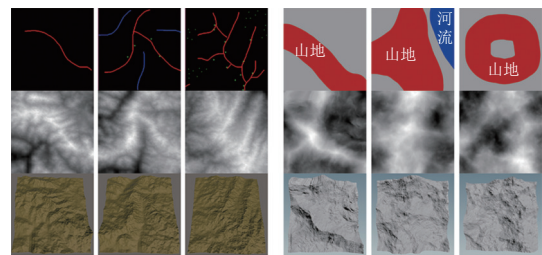


图 12 地形生成软件使用效果

表 1 本文算法与基于过程式的生成方法运行效果

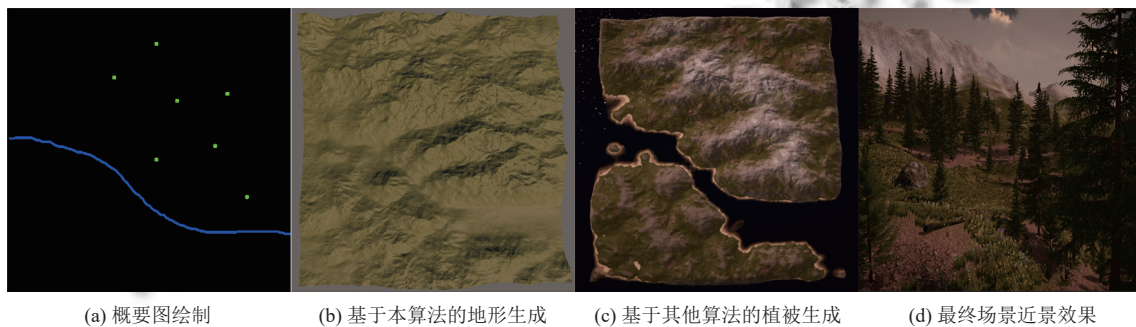
算法	运行环境	电脑配置	平均编辑时间 (s)	<i>TAS</i> 值
本文	Windows 10 Python 3.7	CPU: Core i7-7700HQ; 主 频: 2.80 GHz; 内存: 16 GB	53.2	0.959
对比	Windows 10 Houdini 19.0	CPU: Core i7-7700HQ; 主 频: 2.80 GHz; 内存: 16 GB	622.2	0.936



(a) 本文算法生成效果

(b) 对比实验生成效果

图 13 部分对比实验生成效果



(a) 概要图绘制

(b) 基于本算法的地形生成

(c) 基于其他算法的植被生成

(d) 最终场景近景效果

图 14 使用本算法进行森林场景生成的部分流程及效果

6 结论与展望

本文结合深度学习方法 and 传统图像处理方法对地形生成算法进行了探究,提出了基于概要图的地形生成技术,使地形编辑方式更加直观,且效率大幅提升.本文提出的多尺度的概要图提取算法,使数据集更加接近用户手绘效果,让训练得到的模型具有更强的可

用性和泛化能力,实验表明该方法得到的模型可以接受不同稀疏度的手绘概要图输入,并生成完整且合理的地形高度图.本文提出的多尺度的地形细节填充算法很好地弥补了神经网络模型所生成的地形细节不足的问题,并得到更美观逼真的地形模型.

在此研究基础上,未来研究工作将包括以下几个

方面:

1) 地形类型自定义. 本文所提算法只能生成一种地貌类型的地形, 后续尝试设计带有地形类型约束条件的生成模型, 使一张概要图可以生成多种地貌类型的地形;

2) 地形尺寸自定义. 当前算法所生成的地形大小是固定的, 并且只对中等大小的地形生成效果较好, 未来将尝试实现地形尺寸的自定义, 使大型地形也能被合理地生成.

参考文献

- Galín E, Guérin E, Peytavie A, *et al.* A review of digital terrain modeling. *Computer Graphics Forum*, 2019, 38(2): 553–577. [doi: [10.1111/cgf.13657](https://doi.org/10.1111/cgf.13657)]
- Fournier A, Fussell D, Carpenter L. Computer rendering of stochastic models. *Communications of the ACM*, 1982, 25(6): 371–384. [doi: [10.1145/358523.358553](https://doi.org/10.1145/358523.358553)]
- Parberry I. Modeling real-world terrain with exponentially distributed noise. *Journal of Computer Graphics Techniques*, 2015, 4(2): 1–9.
- de Carpentier GJP, Bidarra R. Interactive GPU-based procedural heightfield brushes. *Proceedings of the 4th International Conference on Foundations of Digital Games*. Orlando: ACM, 2009. 55–62.
- Hnaidi H, Guérin E, Akkouche S, *et al.* Feature based terrain generation using diffusion equation. *Computer Graphics Forum*, 2010, 29(7): 2179–2186. [doi: [10.1111/j.1467-8659.2010.01806.x](https://doi.org/10.1111/j.1467-8659.2010.01806.x)]
- Křištof P, Beneš B, Křivánek J, *et al.* Hydraulic erosion using smoothed particle hydrodynamics. *Computer Graphics Forum*, 2009, 28(2): 219–228. [doi: [10.1111/j.1467-8659.2009.01361.x](https://doi.org/10.1111/j.1467-8659.2009.01361.x)]
- Musgrave FK, Kolb CE, Mace RS. The synthesis and rendering of eroded fractal terrains. *ACM SIGGRAPH Computer Graphics*, 1989, 23(3): 41–50. [doi: [10.1145/74334.74337](https://doi.org/10.1145/74334.74337)]
- Cordonnier G, Galin E, Gain J, *et al.* Authoring landscapes by combining ecosystem and terrain erosion simulation. *ACM Transactions on Graphics*, 2017, 36(4): 1–12.
- Zhou H, Sun J, Turk G, *et al.* Terrain synthesis from digital elevation models. *IEEE Transactions on Visualization and Computer Graphics*, 2007, 13(4): 834–848. [doi: [10.1109/TVCG.2007.1027](https://doi.org/10.1109/TVCG.2007.1027)]
- Isola P, Zhu JY, Zhou TH, *et al.* Image-to-image translation with conditional adversarial networks. *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu: IEEE, 2017. 5967–5976.
- Guérin É, Digne J, Galin É, *et al.* Interactive example-based terrain authoring with conditional generative adversarial networks. *ACM Transactions on Graphics*, 2017, 36(6): 228.
- 董婷婷. 基于 DEM 的清河流域大中小型河流特征值自动提取. *水利规划与设计*, 2014, (10): 26–29. [doi: [10.3969/j.issn.1672-2469.2014.10.010](https://doi.org/10.3969/j.issn.1672-2469.2014.10.010)]
- Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. *Proceedings of the 18th International Conference on Medical Image Computing and Computer-assisted Intervention*. Munich: Springer, 2015. 234–241.
- Génevaux JD, Galin E, Peytavie A, *et al.* Terrain modelling from feature primitives. *Computer Graphics Forum*, 2015, 34(6): 198–210. [doi: [10.1111/cgf.12530](https://doi.org/10.1111/cgf.12530)]

(校对责编: 牛欣悦)