

基于 OpenGL 与 Perlin 噪声的 3D 地形仿真^①



黄泽华¹, 铁治欣^{1,2}, 陈 强¹

¹(浙江理工大学 信息学院, 杭州 310018)

²(浙江理工大学科技与艺术学院, 绍兴 312369)

通信作者: 铁治欣, E-mail: tiezx@zstu.edu.cn

摘 要: 随着互联网技术的革新与进步, 3D 仿真技术在游戏、动漫中的应用愈发显得重要. 针对随机中点位移法生成的山体缺乏较好的细节性以及失真性问题, 本文提出一种基于 OpenGL 与 Perlin 噪声算法生成仿真 3D 地形环境的方法. 传统 Perlin 噪声算法生成的高程图较为平缓, 因此采用分形及湍流算法对高程图进行优化, 可以产生明显冲突, 更符合山峰的特征. 针对地形图像贴图过程中出现的过渡断层现象, 采用分层采样策略, 使图像呈现更加自然. 在光照添加的过程中, 提出了多次二分平均算法计算平面法线向量, 在山体光线的呈现上有更加圆滑的效果. 实验结果表明, 该方法可以很好地创建 3D 地形的三维网格, 同时能够表现出良好的仿真效果.

关键词: OpenGL; Perlin 噪声; 地形仿真

引用格式: 黄泽华, 铁治欣, 陈强. 基于 OpenGL 与 Perlin 噪声的 3D 地形仿真. 计算机系统应用, 2022, 31(1): 182-189. <http://www.c-s-a.org.cn/1003-3254/8264.html>

3D Terrain Simulation Based on OpenGL and Perlin Noise

HUANG Ze-Hua¹, TIE Zhi-Xin^{1,2}, CHEN Qiang¹

¹(School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China)

²(Keyi College of Zhejiang Sci-Tech University, Shaoxing 312369, China)

Abstract: With the innovation and progress of Internet technologies, the application of three-dimensional (3D) simulation technologies in games and animations has become increasingly important. In response to the lack of detail and the distortion of the mountains generated by the random midpoint displacement method, this paper proposes a method of generating simulated 3D terrain environments based on the OpenGL technology and the Perlin noise algorithm. The elevation maps generated by the traditional Perlin noise algorithm are relatively flat. Therefore, they are optimized with the fractal and turbulence algorithms, which produce obvious conflicts that are more in line with the characteristics of the mountains. For the transitional fault phenomena in the process of terrain image mapping, a layered sampling strategy is adopted to make the image more natural. In the case of added light, the plane normal vector is obtained through the multiple binary averaging algorithm, which delivers a smoother mountain light representation. The experimental results show that the method not only realizes the 3D grid construction of 3D terrain but also shows a favorable simulation effect.

Key words: OpenGL; Perlin noise; terrain simulation

自然环境的展现在许多应用中是必不可少的, 如地理信息系统^[1], 飞行模拟或视频游戏等. 地形是这些室外环境的关键部分, 其表示方式已在文献中得到广

泛分析^[2,3]. 一些研究试图提取真实数据^[4], 但结果受到真实世界的限制, 无法总是能够获得所有真实数据来表示它们. 因此自动地形生成非常重要, 仍是一个活跃

① 收稿时间: 2021-03-18; 修改时间: 2021-04-16; 采用时间: 2021-04-30; csa 在线出版时间: 2021-12-17

的研究领域. 利用 Mandelbrot^[5], Prusinkiewicz 等人^[6]提出的分形维数, 出现了许多研究, 这些研究主要使用过程建模来解决其创建过程的自动化^[7]. 这种建模方法已被用于表示许多与地形有关的特征, 包括道路、树木、村庄、城市景观^[8-10]等, 甚至用于表示大气现象^[11]. 根据 Smelik 等人^[12]的工作, 过程建模最初是基于细分过程, 其中高程图是迭代创建的, 为了创建高程, 在每次迭代中添加了受控量的随机性. 基于过程建模的第一个细分算法称为中点偏移方法^[13,14], 其中每次迭代都会生成一个新点, 该新点的高程是通过其邻居高程的平均值获得的, 并添加了随每次迭代而减小的随机位移. 许多其他研究, 主要是基于噪声发生器, 如 Perlin 方法^[15], 其中高程图是通过湍流函数获得的. Perlin 噪声很快, 而且会创建均匀且随机的地形, 通常需要进行大量的后处理才能添加有趣的特征. 例如 Parberry^[16]提出的一项工作, 它使用 Perlin 噪声的一种变体——值噪声, 根据从地理信息系统 (geographic information system, GIS) 数据中提取的高程数据的空间分析来生成典型的程序地形. 基于噪声发生器的方法的优点在于, 由于每个点都可以独立于其邻域进行计算, 因此计算成本较低.

在国内, 张润花等^[17]利用随机中点位移法绘制地形, 但由于算法随机性, 使网格细分的范围难以选择, 容易产生地形的失真现象; 李玉娟等^[18]利用基于数字高程模型 (digital elevation model, DEM) 的方法, 但该方法数据量较大, 绘制效率较低; 马海凤^[19]对 GeoMip-Mapping 算法进行改进, 利用 LOD 算法绘制地形, 在效率上有了提高, 但未考虑动态地形的情况; 郑顾平等^[20]利用 WebGL 标准进行动态地形绘制, 是一种基于 Web 技术下动态调用 OpenGL 等图形库进行地形绘制, 但只是实现了几何剪切图算法.

综合对上述论文的研究, 本文利用改进的 Perlin 噪声算法生成高程图, 导入 OpenCV 库对图像信息进行读取, 采用网格生成算法生成模型数据, 地形模型以及贴图数据, 在 OpenGL 环境下生成仿真地形图像, 解决了随机中点位移法构建地形中产生的失真及仿真效果较差的问题. 并且优化了光照系统, 在保持其随机性, 低成本计算的同时, 使绘制出的地形网格图细节更为丰富, 成像效果更为真实.

1 Perlin 噪声介绍

图形噪声, 是计算机图形学中一类伪随机算法, 经

常用于模拟自然界中的各种纹理材质, 如云^[21]、波浪^[22]、水面^[23]、湍流^[24]、雪花^[25]、场景渲染^[26,27]等. 通过伪随机算法计算出的伪随机数是属于 $[0, 1]$ 均匀分布的随机数序列, 并不真正的随机, 但具有类似于随机数的统计特征. Perlin^[15]在 1985 年的 SIGGRAPH 大会上首次提出 Perlin 函数. 后又在 2001 年, 提出了 Simplex 噪声算法来减小计算复杂度. 目前已经被广泛应用于计算机仿真领域及三维游戏制作中. Perlin 噪声是一种基于晶格的梯度噪声, 随机数的创建过程包括定义晶格结构, 输入多维点坐标, 使用缓和曲线来计算他们的权重和 3 个过程. 以 Perlin 噪声为基础, 后续利用分形, 湍流等手段, 可以生成多种不同有趣的噪声图.

2 3D 地形仿真的总体设计

本文主要通过 Perlin 噪声生成随机高程图, 以读取高程图单通道颜色值映射为地形高程图为主要思路, 形成三维地形网格图. 在构建三维地形网格图时, 选择以 TRIANGLES 方式^[28]绘制, 因此对于顶点数组的构造方式决定了程序处理的快慢. 假设地图大小为 $n \times m$, 以全顶点绘制需要构建 $(n-1)(m-1) \times 6$ 个顶点, 而以索引方式来构建, 仅需 $n \times m$ 个顶点, 减小了约 83% 的内存占用率, 提高了运行效率. 在仿真视觉成效方面, 对不同高度的位置采用不同的贴图进行覆盖. 在贴图交界处, 以模糊函数进行叠加计算, 使过渡效果更加平滑. 光照效果往往能提高整体的视觉体验, 在最终成像结果中加入光照系统, 以提升对整体细节的优化. 总结各项工作, 形成总体流程图如图 1 所示.

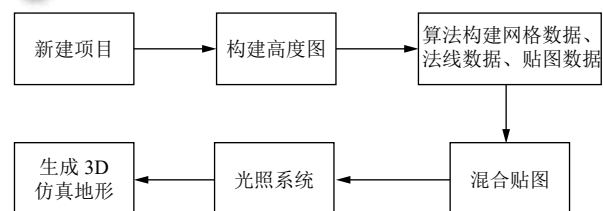


图 1 总体流程图

3 3D 地形仿真的生成

将 3D 地形生成仿真分为高程图的生成与改进、数据集生成、贴图策略设计与光照系统设计几个部分.

3.1 基于柏林噪声的高程图生成

假设要生成一个 $N \times N$ 的高程图, 首先使用式 (1) 与式 (2) 初始化索引表 P 与随机矩阵 G . 式 (1) 中随机

数赋值为 0 到 N 之间无重复任意值, 式 (2) 中随机矩阵用于对晶格进行赋值. 基于随机矩阵的做法, 与 Perlin 于 2002 年的改进算法^[29] 相比, 在计算量方面有所增加, 但能够提高生成噪声图像的随机性.

$$P[2N] = [i_0, i_1, \dots, i_{N-1}, i_0, i_1, \dots, i_{N-1}] \quad (1)$$

其中, $i_j(j=0, 1, 2, \dots, N-1)$ 是 0 到 $N-1$ 之间无重复随机整数.

$$G[2N][3] = \begin{bmatrix} J_{0,0} & J_{0,1} & J_{0,2} \\ J_{1,0} & J_{1,1} & J_{1,2} \\ \dots & \dots & \dots \\ J_{N-1,0} & J_{N-1,1} & J_{N-1,2} \\ J_{0,0} & J_{0,1} & J_{0,2} \\ J_{1,0} & J_{1,1} & J_{1,2} \\ \dots & \dots & \dots \\ J_{N-1,0} & J_{N-1,1} & J_{N-1,2} \end{bmatrix} \quad (2)$$

其中,

$$J_{p,q} = \text{Random}() \% 2N / N - 1, p = 0, 1, 2, \dots, N-1, q = 0, 1, 2.$$

如图 2 所示, 假设输入点 $S(x, y, z)$, 通过该点可计算得到 S_1, S_2, \dots, S_8 , 共 8 个晶格值. 首先通过输入点 x 值, 计算 bx_0, bx_1, rx_0, rx_1 . 其中, $bx_0 = \text{int}(x) \% N, bx_1 =$

$(\text{int}(x) \% N + 1) \% N, rx_0 = (x - \text{int}(x)) \% N, rx_1 = (x - \text{int}(x)) \% N - 1$. 同理可得, $by_0, by_1, ry_0, ry_1, bz_0, bz_1, rz_0, rz_1$.

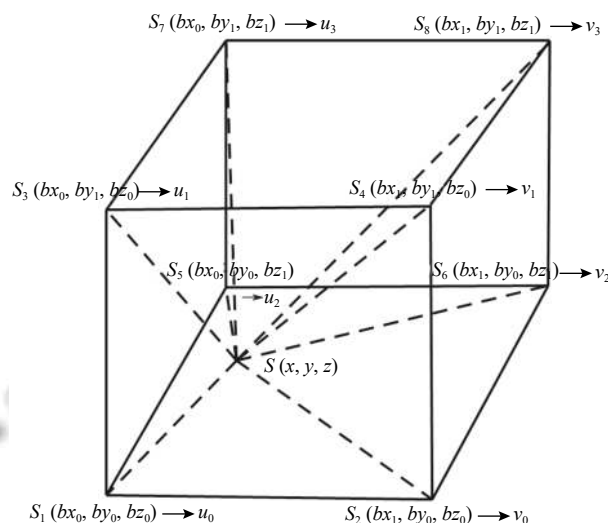


图 2 晶格图示

结合索引表和随机矩阵, 计算出晶格各点的权重值.

$$\begin{bmatrix} u_0 & v_0 \\ u_1 & v_1 \\ u_2 & v_2 \\ u_3 & v_3 \end{bmatrix} = \begin{bmatrix} G'[P[bx_0 + by_0] + bz_0] \\ G'[P[bx_0 + by_1] + bz_0] \\ G'[P[bx_0 + by_0] + bz_1] \\ G'[P[bx_0 + by_1] + bz_1] \end{bmatrix} \begin{bmatrix} rx_0 \\ ry_0 \\ rz_0 \\ rx_1 \\ ry_1 \\ rz_1 \end{bmatrix} \quad (3)$$

其中, $G'[i] = [G[i][0] \quad G[i][1] \quad G[i][2]]$.

以 Perlin 光滑函数^[29] 为权重函数 (4), 以 rx_0, ry_0, rz_0 为参数, 计算得到 $\omega_x, \omega_y, \omega_z$. 通过线性插值函数 (5) 及晶格点的权重值 (3), 得到两个晶格点的线性插值, 如 S_1 与 S_2 的线性插值 a_0 , 如式 (6) 所示. 同理可以得到插值 b_0, a_1, b_1 , 如式 (7)–式 (9) 所示. 最终再对 4 个插值 a_0, b_0, a_1, b_1 再进行二次插值, 可得三维 Perlin 噪声函数, 如式 (10) 所示.

$$\omega(x) = 6x^5 - 15x^4 + 10x^3 \quad (4)$$

$$\text{Lerp}(a, b, c) = (1 - a) \times b + a \times c \quad (5)$$

$$a_0 = \text{Lerp}(\omega_x, u_0, v_0) = (1 - \omega_x)u_0 + \omega_x v_0 \quad (6)$$

$$b_0 = \text{Lerp}(\omega_x, u_1, v_1) = (1 - \omega_x)u_1 + \omega_x v_1 \quad (7)$$

$$a_1 = \text{Lerp}(\omega_x, u_2, v_2) = (1 - \omega_x)u_2 + \omega_x v_2 \quad (8)$$

$$b_1 = \text{Lerp}(\omega_x, u_3, v_3) = (1 - \omega_x)u_3 + \omega_x v_3 \quad (9)$$

$$\begin{aligned} \text{PerlinNoise}(X, Y, Z) &= (1 - \omega_z) \text{Lerp}(\omega_u, a_0, b_0) + \omega_z \text{Lerp}(\omega_y, a_1, b_1) \\ &= (1 - \omega_z) ((1 - \omega_y)a_0 + \omega_y b_0) + \omega_z ((1 - \omega_y)a_1 + \omega_y b_1) \end{aligned} \quad (10)$$

以要生成 $N \times N$ 的高程图中的每个像素点的坐标为 x, y 坐标值, 以当前系统时间为 z 坐标值, 利用 Perlin 噪声函数 (10), 计算图片中每个像素点对应的高度值, 可得基础高程图, 如图 3 所示, 其中取 $N = 1024$.

3.2 高程图改进

经过经典 Perlin 噪声算法得到的高程图冲突效果

较差,可以利用分形(Fbm)与湍流(Turbulence)算法进行改进.分形布朗运动算法主要对不同频率和振幅的噪声函数进行叠加,如式(11)所示.

$$Fbm = \sum_i PerlinNoise(F_i \times (X, Y, Z)) \times A_i \quad (11)$$

$$F_{i+1} = F_i \times L \quad (12)$$

$$A_{i+1} = A_i \times G \quad (13)$$

其中, A 表示每一次噪声叠加的权值, F 表示叠加噪声的比例, A 与 F 的初始值为 1. i 为循环次数, 表示多张高程图的叠加效果, 通过测试, 令 i 为 4 可取得较好的效果. L 和 G 表示每次迭代中修改 A 和 F 值的乘系数, 分别赋值为 2 与 1/2, 使每次叠加的噪声权值和大小不同. 最终得到高程图如图 4 所示.

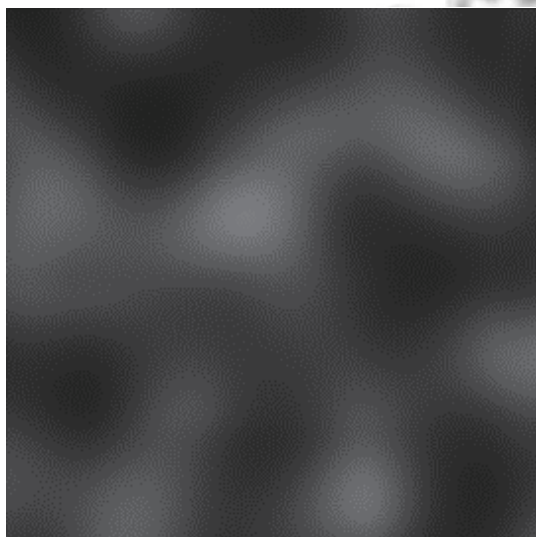


图3 Perlin 基础高程图

在利用 Fbm 进行改进的基础上, 运用式(14)进行 Turbulence 计算, Turbulence 对噪声函数取绝对值, 使噪声值等于 0 处发生突变, 产生湍流纹理, 最终得到高程图如图 5 所示.

$$Turbulence = \sum_i |PerlinNoise(F_i \times (X, Y, Z))| \times A_i \quad (14)$$

其中, 字符含义与式(11)一致.

为验证算法的有效性, 在相同条件下, 与相晓嘉等^[30]提出的基于 Perlin 噪声的高分辨率地形仿真方法(PNB_HRTDS 方法)进行比较, 结果如图 6、图 7、表 1 所示. 图 6 与图 7 分别是本文方法与 PNB_HRTDS 方法生成的大小为 1024×1024 的地形仿真图. 两图相比, 可以看出: PNB_HRTDS 方法生成的地形仿真图的效果

过于平整, 山体表面褶皱缺少, 山体之间的细节不够充分. 由表 1 可知, 在不同大小高程图中, 本文方法的高程值的均值有较大提升, 且标准差较大, 说明高程图中冲突效果明显, 山体的表现效果有所提升.

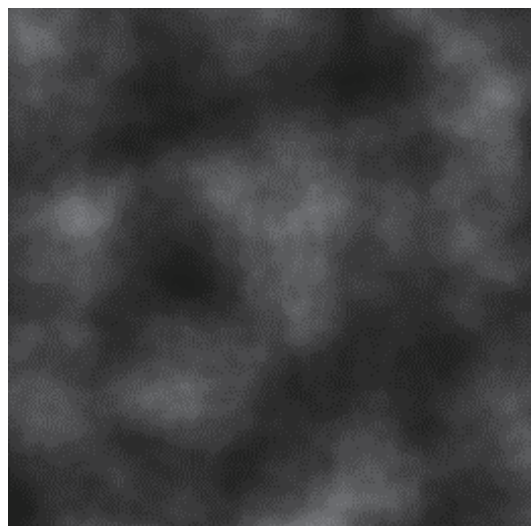


图4 Fbm 高程图

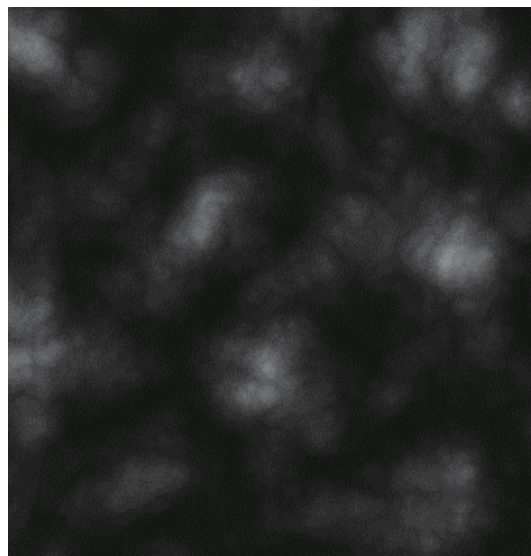


图5 Turbulence 高程图

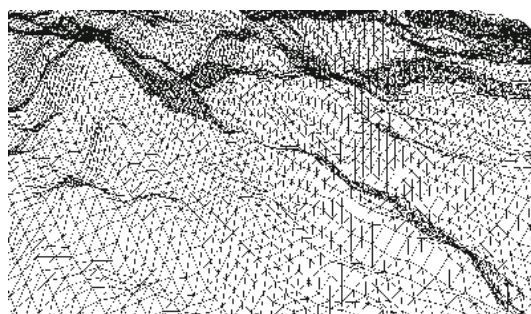


图6 本文方法地形仿真效果图

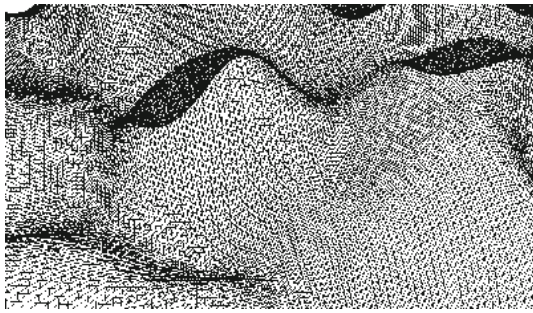


图7 PNB_HRTDS方法地形仿真效果图

假设要生成 $n \times n$ 大小的高程图,两种算法均需申请大小为 n 的float类型数组作为索引表 P 与大小为 $6n$ 的float类型数组作为随机矩阵 G ,即需要 $7n$ 个float类型的存储空间,所以两种算法的空间复杂度为均 $O(n)$ 。对于 $n \times n$ 大小的高程图中的每一点,PNB_HRTDS方法均需要调用Perlin噪声函数一次,生成该点的高程,所以其时间复杂度为 $O(n^2)$;本文方法生成一个点的高程需要调用4次Perlin噪声函数,所以其时间复杂度也为 $O(n^2)$,但实验生成时间要比PNB_HRTDS方法慢一点。

表1 生成数据对比

方法	高程图大小 ($N \times N$)	高程均值	高程标准差	生成时间 (ms)	所需存储空间 (KB)
本文	1 024	0.605 3	0.219 4	334	9 227
	1 536	0.605 2	0.219 3	644	19 823
	2 048	0.605 3	0.219 5	1 368	33 160
	2 560	0.605 2	0.219 4	2 141	51 888
	3 072	0.605 3	0.219 3	2 578	74 208
PNB_HRTDS	1 024	0.357 4	0.165 1	77	9 227
	1 536	0.357 1	0.165 1	194	19 823
	2 048	0.357 3	0.165 1	318	33 160
	2 560	0.357 1	0.165 1	503	51 888
	3 072	0.357 1	0.165 1	585	74 208

3.3 数据集生成

3.3.1 网格数据生成

通过OpenCV库,对前述所生成的高程图以单通道模式进行读取,存入图像信息矩阵 A 中.以像素点颜色值映射为高度值,生成方式如图8所示。

在图8中,Width、Height表示高程图的宽度与高度,vertices表示网格数组, I 、 J 、 K 表示宽高度与数组的递增索引, A_{ij} 表示高程图中读出的高度值,此外还加入了vertexScale、yScale,用于控制地图大小的缩放比例与地图高度的缩放比例。

在网格数组的基础上,可以使用索引的方式,优化网格数据结构,加快运行效率,建立与顶点数据对应的索引数组indices,如图9所示,字符含义与图8一致。

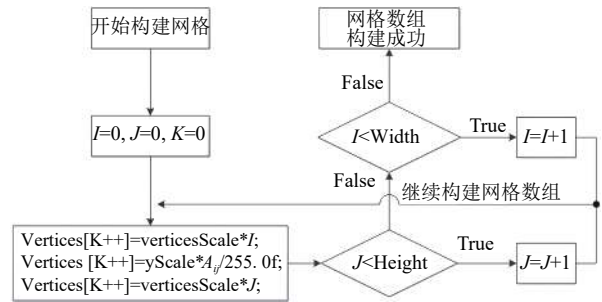


图8 网格数据生成

3.3.2 法线数据生成与优化

OpenGL以TRIANGLES方式绘制图形,三角形是绘制的最小图元,故可基于三角形的数据,计算垂直于该平面的垂直向量.一般情况下通过三角图元的各顶点位置信息单独计算,如刘鹏等^[31]进行的隧道仿真系统的研究与开发,使用glNormal3fv()函数进行法线计算.在本文的计算方法中,由于网格数组的优化,同一顶点复用,会产生单个顶点包含多个垂直向量问题.本文以多次二分平均算法优化该问题,使垂直单位向量变化更加平滑。

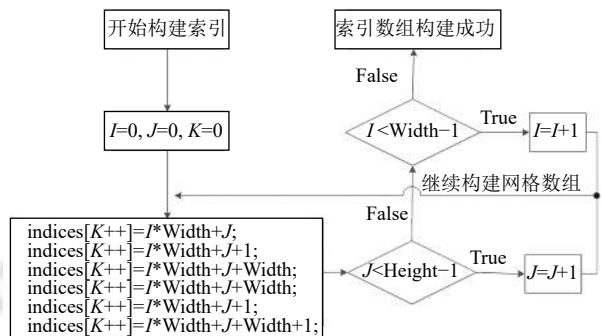


图9 索引数据生成

如图10(a)所示,存在两个三角图元ABD与ACD,A、D两点处是索引情况下的复用点.虚线AF是ABD图元中顶点A的法线,虚线AE是ACD图元中A点的法线.此处以A点为例,由于A点是复用顶点,只能存在唯一法线.在不复用情况下,每一顶点都属于唯一图元,如图10(b)所示,图元中任意处的法线都与顶点一致.使用多次二分平均法处理后,使用AE与AF的平分线AG作为A点的法线,作为解决冲突的手段.在顶点被多次复用的情况下,进行多次二分即可.在法线修正后,如图10(c)所示,A点的法线由AF替换为AG,从而AB线段中的位置,也会依据比例进行调整,如

AB 中点 H, 法线的方向指向 A 点与 B 点法线方向的折中方向。

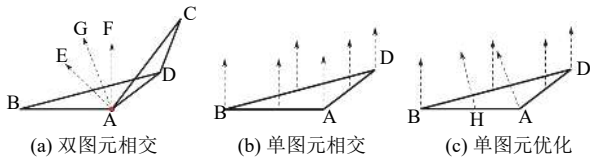


图 10 法线优化

根据本文提出优化方案, 与刘鹏程等^[31]方案中使用 `glNormal3fv()` 函数进行法线计算进行对比, 图 11(a) 为本文方案, 图 11(b) 为常规方案。从图中可知, 在使用光照计算后, 在最小图元的交界处的左右两侧产生亮度的跳跃性变化。而本文优化方案下, 不仅保持了光线的漫反射效果, 并且在山体光线的呈现上有更加圆滑的效果。

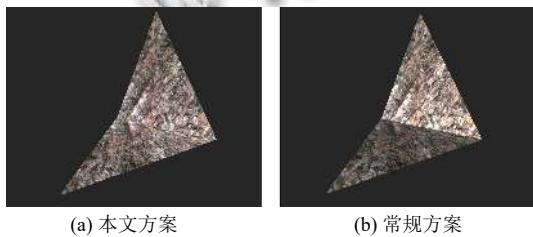


图 11 法线优化效果对比

3.4 贴图策略设计

数据集构建完成后, 可以依照网格数组来生成三维地形的网格模型。但单纯的模型只能反映山体的地形结构, 不能模拟真实的仿真效果。因此可以通过三维纹理贴图方法增强仿真效果。三维纹理贴图指将二维纹理图像张贴到三维模型的表面上, 从而得到逼真的三维图形的过程。然而, 对于复杂三维模型的纹理贴图还没有找到直接贴图的方法^[32]。通常情况下会使用一种适合场景的纹理进行贴图, 如图 12 所示, 但效果会显得较为单一。本文依据对空间高度的划分, 使用雪、岩石、草坪 3 种图像进行分层采样。将高度以一定比例划分为 3 个区域, 在该区域内, 使用对应的图像进行贴图: 如图 13 所示, 在红色框标示区域, 不同层次交界处有较强的截断感。可以使用平滑函数在不同层次之间增加渐变层进行优化, 使效果更为平滑: 如图 14 所示, 红色框标示区域内, 渐变层的仿真效果更为真实, 整个山体的层次更加自然。



图 12 单纹理贴图

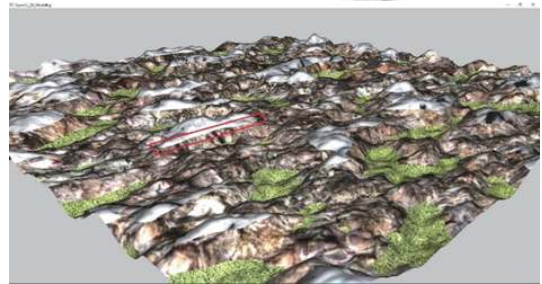


图 13 多纹理贴图

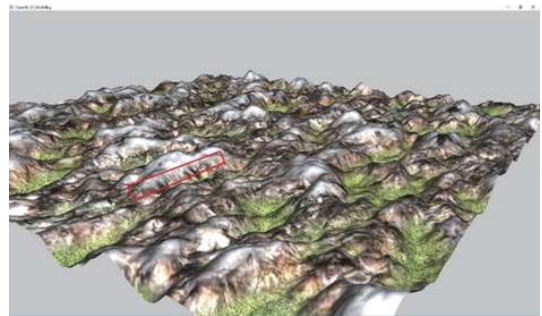


图 14 采用分层采样策略的多纹理贴图

3.5 光照系统设计

现实世界的光照是极其复杂的, 而且会受到诸多因素的影响, 这是我们有限的计算能力所无法模拟的。因此本文对光照使用冯氏光照模型 (Phong lighting model)^[33] 对整体进行光照添加。冯氏光照模型的主要结构由 3 个分量组成: 环境 (ambient)、漫反射 (diffuse) 和镜面 (specular) 光照。环境分量主要取决于物体的颜色, 以一定亮度百分比进行模拟。漫反射与镜面光照是产生光线效果的关键, 设计如图 15 展示。在漫反射中, 如图 15(a) 所示, \vec{N} 是法向量, \vec{L} 是光源方向向量, 将两向量化为单位向量之后, 进行点乘运算, 得到 0~1 之间的数值 A , 为了防止光线从后背穿过, 则取 A 与 0 之间的最大值, 该值表示漫反射的强弱程度, 越接近 1, 则越接近物体原来的颜色。在镜面反射中, 如图 15(b) 所示, \vec{N} 是法向量, \vec{L} 是光源方向向量, \vec{M} 是 \vec{L} 经过平面反射之

后的射线, \vec{K} 为视线的方向向量. 将 \vec{M} 与 \vec{K} 化为单位向量之后, 进行点乘运算, 得到 0~1 之间的数值 B , 其含义与 A 一致. 最终将物体色值的较低百分比如 10%, 及 A 、 B 与色值的乘积相加, 得到物体最终在某一视角中应该呈现的颜色.

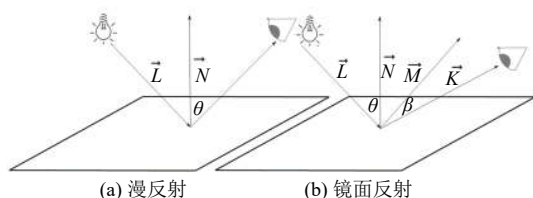


图 15 漫反射及镜面反射光照示意图



图 16 本文方法地形仿真效果图

4 实验结果与分析

本文方法实现的硬件环境为: Intel(R) Core(TM) i7-7700HQ CPU @ 2.80 GHz, 16.00 GB 内存, NVIDIA GeForce GTX 1050 Ti 显卡, Win10 64 位操作系统, Microsoft Visual Studio15 编译器. 本文在通过 Perlin 噪声生成高程图的基础上, 构造了网格数组, 优化了法线计算. 实验程序的运行结果如图 14、图 16 所示. 从图 14 中观察到, 在不同的高纬度, 有良好的渐变效果, 雪, 岩石, 草坪之间过渡的比较自然. 图 16 为光源置于图片右侧较高处, 模拟自然界阳光的照射情况, 可以看到山体背阳面有明显的阴影效果, 向阳面有明显的光照效果, 呈现出更贴近现实的仿真效果, 没有过于尖凸的现象出现, 而且山体表面褶皱也表现较好.

5 结束语

本文的工作主要是提出了一种基于 OpenGL 与 Perlin 噪声算法来生成不同的随机地形的办法. 在 Perlin 噪声函数的基础上, 以分形布朗运动为原型, 对噪声进行叠加, 再对冲突部分进行湍流的计算, 最终得到的高程图没有跳跃性与突变型, 显示了良好的效果. 此外, 该方法保持了 Perlin 噪声的随机性质, 每次运行都会产生不同的高程图. 通过解析高程图来获得点位的高度信息, 并且导入顶点数组中, 并以索引的方式减少了数据冗余和加快了渲染效率. 最后通过多次二分平均算法来计算法线, 结合贴图策略及光照系统完成了 3D 地形仿真的整体内容. 实验结果表明, 该方法能够创建完整的 3D 地形, 并且呈现出良好的仿真效果. 本文方法同样适应于火焰, 云朵等的仿真生成.

参考文献

- 1 Rebollo C, Remolar I, Gumbau J, *et al.* Three-dimensional trees for virtual globes. *International Journal of Digital Earth*, 2014, 7(10): 789–810. [doi: 10.1080/17538947.2013.783126]
- 2 Ramos F, Chover M, Ripolles O, *et al.* Continuous level of detail on graphics hardware. *Proceedings of the 13th International Conference on Discrete Geometry for Computer Imagery*. Szeged: Springer, 2006. 460–469.
- 3 Ripolles O, Ramos F, Puig-Centelles A, *et al.* Real-time tessellation of terrain on graphics hardware. *Computers & Geosciences*, 2012, 41: 147–155.
- 4 Li WH, Han DF, Li HY, *et al.* Extraction of digital terrain model based on regular mesh generation in mountainous areas. *Multimedia Tools and Applications*, 2018, 77(5): 6267–6286. [doi: 10.1007/s11042-017-4535-y]
- 5 Mandelbrot BB, Wheeler JA. The fractal geometry of nature. *American Journal of Physics*, 1983, 51(3): 286–287. [doi: 10.1119/1.13295]
- 6 Prusinkiewicz P, Hammel M. A fractal model of mountains and rivers. *Graphics Interface*. Canadian Information Processing Society, 1993. 174–180.
- 7 Rebollo C, Remolar I, Chover M, *et al.* A comparison of multiresolution modelling in real-time terrain visualisation. In: Laganá A, Gavrilova ML, Kumar V, *et al.*, eds. *Computational Science and Its Applications*. Berlin Heidelberg: Springer, 2004. 703–712.
- 8 Emilien A, Bernhardt A, Peytavie A, *et al.* Procedural generation of villages on arbitrary terrains. *The Visual Computer*, 2012, 28(6–8): 809–818. [doi: 10.1007/s00371-012-0699-7]
- 9 Hou F, Qin H, Qi Y. Procedure-based component and architecture modeling from a single image. *The Visual*

- Computer, 2016, 32(2): 151–166. [doi: [10.1007/s00371-015-1061-7](https://doi.org/10.1007/s00371-015-1061-7)]
- 10 Pajarola R, Gobbetti E. Survey of semi-regular multiresolution models for interactive terrain rendering. *The Visual Computer*, 2007, 23(8): 583–605.
- 11 Puig-Centelles A, Ripolles O, Chover M. Creation and control of rain in virtual environments. *The Visual Computer*, 2009, 25(11): 1037–1052. [doi: [10.1007/s00371-009-0366-9](https://doi.org/10.1007/s00371-009-0366-9)]
- 12 Smelik RM, Tutenel T, Bidarra R, *et al.* A survey on procedural modelling for virtual worlds. *Computer Graphics Forum*, 2014, 33(6): 31–50. [doi: [10.1111/cgf.12276](https://doi.org/10.1111/cgf.12276)]
- 13 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)]
- 14 Miller GSP. The definition and rendering of terrain maps. *ACM SIGGRAPH Computer Graphics*, 1986, 20(4): 39–48. [doi: [10.1145/15886.15890](https://doi.org/10.1145/15886.15890)]
- 15 Perlin K. An image synthesizer. *ACM SIGGRAPH Computer Graphics*, 1985, 19(3): 287–296. [doi: [10.1145/325165.325247](https://doi.org/10.1145/325165.325247)]
- 16 Parberry I. Designer worlds: Procedural generation of infinite terrain from real-world elevation data. *Journal of Computer Graphics Techniques*, 2014, 3(1): 74–85.
- 17 张润花, 刘树群, 赵付青. 地形生成的细分随机移位方法. *华中师范大学学报(自然科学版)*, 2012, 46(5): 533–536.
- 18 李玉娟, 谭同德. 三维场景中大规模地形地貌的生成. *计算机应用与软件*, 2013, 30(11): 131–135. [doi: [10.3969/j.issn.1000-386x.2013.11.037](https://doi.org/10.3969/j.issn.1000-386x.2013.11.037)]
- 19 马海凤. 大规模地形可视化技术研究 [硕士学位论文]. 西安: 西安电子科技大学, 2015.
- 20 郑顾平, 白若林. 基于 WebGL 的动态地形实时绘制. *软件导刊*, 2017, 16(12): 202–204, 209.
- 21 王帅, 纪玉波. 基于 Simplex 噪声绘制云的方法. *辽宁石油化工大学学报*, 2009, 29(3): 58–61. [doi: [10.3969/j.issn.1672-6952.2009.03.017](https://doi.org/10.3969/j.issn.1672-6952.2009.03.017)]
- 22 石秋华, 孙娟. 利用 Perlin 噪声生成水波面的动态模拟研究. *软件导刊*, 2012, 11(2): 140–142.
- 23 项予, 许森. 基于 Perlin 噪声的动态水面实时渲染. *计算机工程与设计*, 2013, 34(11): 3966–3970. [doi: [10.3969/j.issn.1000-7024.2013.11.049](https://doi.org/10.3969/j.issn.1000-7024.2013.11.049)]
- 24 Bridson R, Houriham J, Nordenstam M. Curl-noise for procedural fluid flow. *ACM Transactions on Graphics*, 2007, 26(3): 46–es. [doi: [10.1145/1276377.1276435](https://doi.org/10.1145/1276377.1276435)]
- 25 徐建华, 顾浩, 康凤举, 等. 真实感降雪场景可视化. *系统仿真学报*, 2014, 26(10): 2391–2394, 2399.
- 26 Barnard R, Ural S. Rendering translucency with Perlin noise. *Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*. New Zealand: ACM, 2005. 131–134.
- 27 Coninx A, Bonneau GP, Droulez J, *et al.* Visualization of uncertain scalar data fields using color scales and perceptually adapted noise. *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*. Toulouse: ACM, 2011. 59–66.
- 28 施志辉, 徐哲, 葛研军. 基于 OpenGL 的三角形网格切屑仿真的实现. *机械设计与制造*, 2003, (1): 73–75. [doi: [10.3969/j.issn.1001-3997.2003.01.035](https://doi.org/10.3969/j.issn.1001-3997.2003.01.035)]
- 29 Perlin K. Improving noise. *ACM Transactions on Graphics*, 2002, 21(3): 681–682. [doi: [10.1145/566654.566636](https://doi.org/10.1145/566654.566636)]
- 30 相晓嘉, 朱华勇. 基于 Perlin 噪声的高分辨率地形仿真方法. *计算机仿真*, 2007, 24(12): 174–177. [doi: [10.3969/j.issn.1006-9348.2007.12.046](https://doi.org/10.3969/j.issn.1006-9348.2007.12.046)]
- 31 刘鹏程, 艾廷华, 胡晋山. 基于 OpenGL 技术的隧道仿真系统的研究与开发. *计算机工程*, 2008, 34(6): 240–242. [doi: [10.3969/j.issn.1000-3428.2008.06.087](https://doi.org/10.3969/j.issn.1000-3428.2008.06.087)]
- 32 王波, 孙蔚. 基于 OpenGL 的新式 OBJ 文件纹理贴图方法研究. *计算机与数字工程*, 2015, 43(8): 1497–1500. [doi: [10.3969/j.issn1672-9722.2015.08.029](https://doi.org/10.3969/j.issn1672-9722.2015.08.029)]
- 33 Phong BT. Illumination for computer generated pictures. *Communications of the ACM*, 1975, 18(6): 311–317. [doi: [10.1145/360825.360839](https://doi.org/10.1145/360825.360839)]