Scientific Journal of Impact Factor(SJIF): 3.134

e-ISSN(O): 2348-4470
p-ISSN(P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 2,Issue 5, May -2015

Multiple Hill Terrain Generation by Displacing Two Dimensional Grid Points

Chetan K. Sindhi

Narsinhbhai Institute of Computer Studies and Management, S V Campus, Kadi – 382715, Gujarat, India

Abstract: *Terrain modeling plays an important role in computer graphics. There are several methods to generate terrain models. Displacing two-dimensional grid points provide a simple way to generate realistic landscape. The basic idea here is to generate a continuous surface which varies in height randomly but with the random variation obeying a particular statistical law.*

The method here used to generate a Terrain is a simplified form of Midpoint Displacement method. The two dimensional grid of X-Y coordinates is displaced and then the height values are assigned to each point on two dimensional grid. The height values can be assigned in a way to control the shape of the terrain. Here two algorithms are presented: The first one is intended to generate a single hill on the two dimensional grid and the second one is to generate multiple hills.

Keywords: *Terrain, Two-Dimensional Grid, Displacement, Single Hill Terrain, Multiple Hill Terrain*

I. INTRODUCTION

One of the biggest problems of geometry is to describe the shape of a cloud, a tree or a mountain. The reason is that the shapes of these objects are irregular and cannot be analyzed in standard forms. This means that a cloud cannot be represented as a sphere or ellipsoid nor a mountain as a cone. Therefore, we can say that in nature many patterns (actually the most of them) are irregular and fragmented.

Terrain is used as a general term in geography, especially physical geography, referring to the lie of the land. Terrain is an area of the surface (landscape) with distinctive geological characteristics.

II. LITERATURE REVIEW

Several researchers have addressed the terrain generation problem. Donald Hearn and M. Paulin Baker et al. [1] use the concept of random midpoint displacement methods, similar to the random displacement methods used in geometric constructions, have been developed to approximate fractional Brownian-motion representations for terrain and other natural phenomena. The midpoint displacement of a line segment is used to generate a two dimensional object. The concept is extended and applied to the two dimensional object in order to generate three dimensional object that look like a terrain. This method is faster than fractional Brownian-motion method; they produce less realistic-looking terrain features.

Similar approach is adopted by Miller et al. [2]. He examines three methods, two existing and one new, for the generation of fractals based on recursive subdivision. Both existing methods are found to have defects as they are based on triangular edge subdivision, which are not present in the new method based on diamond-square subdivision.

Robot Frog et al. [3] developed hill algorithm to generate hills on smooth and regular surface. This algorithm is an iterative procedure and possesses the properties of fractals. A hill is kind of a rounded hump shape is generated by selecting random point on or near the terrain and selecting random radius between point and smooth surface. The number of repetitions chosen will affect the appearance of the terrain. Again this also generates less realistic-looking terrain.

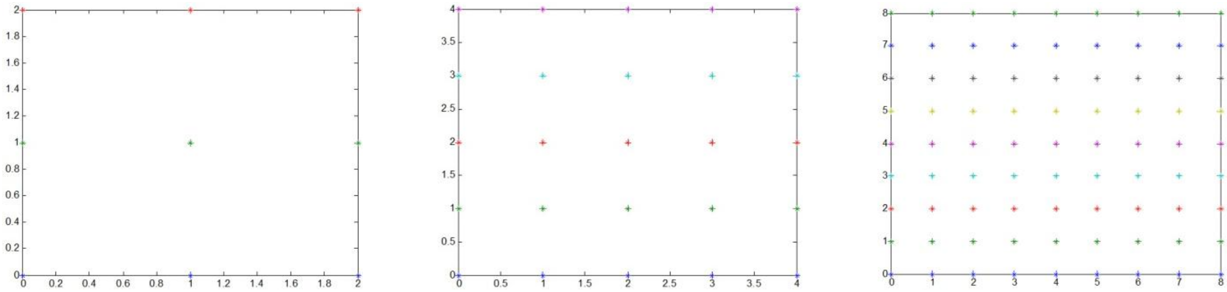
Single hill fractal terrain generation approach discussed in [4] is further extended here to generate multiple hills.

III. OBJECTIVE

- The objective of this paper is to present a simple random terrain generation algorithm.
- The form of terrain is appropriate for visualization applications.
- The shape of the terrain can be controlled to generate single or multiple hills.

IV. DISPLACING TWO - DIMENSIONAL GRID POINTS

The task begin with generating a two dimensional grid of x and y co-ordinates using. The two-dimensional rectangular can be generated by replicating two vectors x and y. The size of two-dimensional grid is determined by the length of the grid vectors. Each of the grid point represents a point in X-Y plane. Few examples of two-dimensional rectangular grid are shown in figure-1.



3 x 3 ***5 x 5*** ***9 x 9***

Figure-1: Two-Dimensional Rectangular Grid of x-y Points

When we consider 3 x 3 grid, it has one midpoint which divides the grid into four equal sub rectangular grids. Similarly for 5 x 5 grid the midpoint divides it into four 3x3 sub grids, which can be further divided by their midpoints. For 9 x 9 grid, we get sixteen 3 x 3 sub grids. We keep track of all these midpoints of 3 x 3 sub grids for Multiple Hill Terrain generation.

Now the grid points are then relocated by adding a random offsets within the range less than the distance between any two grid points. The resultant grid is now not rectangular one. More the number of elements in x and y vectors, more the number of points on the terrain and therefore we can get a terrain that looks more realistic as the points are closer. Figure-2 shows one such displaced two-dimensional grid of points.

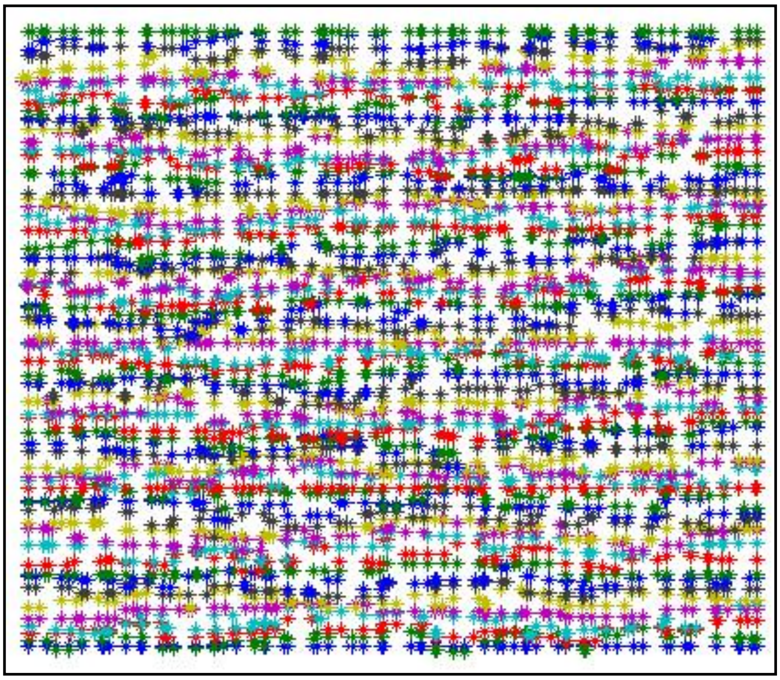


Figure-2: Displaced Two-Dimensional Grid

IV. SINGLE HILL TERRAIN

Now the next task is to generate the height values z. The height values z can be generated in variety of ways. One of the simplest ways is that the original grid of points (x, y) can be considered without any displacement and then the displaced values can be assigned to height value z (as a function of displaced (x, y) value). Initially it seems to be a good approach but it does not give any variety in the sense that the appearance of the surface doesn't look realistic. This is because the grid points (x, y) without any displacement do not appear natural since they are well arranged (not random).

Another approach to find the height values z is as below:
The two co-ordinates x and y specify the position of each point on terrain and the third co-ordinate z specify the height of the terrain at each point (x, y). The third co-ordinate z is then assigned in order to control the shape of the terrain. The height value z is assigned in a way that it takes maximum value at the displaced center point and as it goes towards sides of the terrain the height value z monotonically decreases. The four corners are assigned minimum height, say 0. The three dimensional surface generated by grid points (x, y, z) looks like a single hill mountain. The number of hills on the surface can be controlled by height value z.

Single Hill Terrain Generation Algorithm-1

The stepwise procedure for this approach can be described as follows:

1. Initialize the two dimensional grid (x, y) by replicating x and y vectors.
2. Displace two dimensional grid points (x, y) by adding random offset in the range less than the distance between any two grid points.
3. Assign height value z to each grid point in such a way that it takes maximum value at the displaced center point and as it goes towards sides of the terrain the height value z monotonically decreases.
4. Generate a surface on using this three dimensional grid (x, y, z).

VI. MULTIPLE HILL TERRAIN

The midpoint of the two dimensional grid is given the maximum height. Then in the subsequent steps, the height of the new generated midpoints for 3 x 3 sub grids decreases. For each 3 x x sub grid, the four corner points of sub grids are assigned height 0 and the height of the midpoint is in proportion to its minimum distance from the four corner points of sub grid. So for each sub grid we get a small hill at its midpoint. Thus the Multiple Hill Terrain Algorithm can be stated as below:

Multiple Hill Terrain Generation Algorithm-2

The stepwise procedure for this approach can be described as follows:

1. Initialize the two dimensional grid (x, y) by replicating x and y vectors.
2. Displace two dimensional grid points (x, y) by adding random offset in the range less than the distance between any two grid points.
3. The grid is divided into 3 x 3 sub grids by midpoints.
4. Assign height value $z=0$ to each corner point of 3 x 3 sub grids point and height z of sub grid midpoints in proportion to its minimum distance from the four corner points of sub grid.
5. Generate a surface on using this three dimensional grid (x, y, z).

VII. DISCUSSION

The terrain data obtained by above method doesn't give smoother surface. The terrain data can be made smoother by some interpolation method and then better surface can be fit for the interpolated data.

Matlab provides an in-built surface fitting tool with many surface fitting techniques. Surface Fitting Tool provides a flexible and intuitive graphical user interface where one can interactively fit surfaces to data and view plots.

VIII. RESULTS

The *(x, y, z)* matrix grid obtained by above procedure can be plotted realistically using different surface generation routines available in scientific programming tools like Matlab, Scilab, Python, etc. Here the algorithm is implemented in Matlab. The matrix *(x, y, z)* is plotted using the surface generating function *'surf'* in MATLAB. The *surf* function gives surface with different color intensity values. The colors of the surface are assigned according to the height value by the surf function. The picture with many colors does not look realistic.

To make the terrain more realistic a variant of the function surf, the function *surfl* is used. The *surfl* produces a 3-D shaded surface with lighting. The shading is based on a combination of diffuse, specular and ambient lighting models. The shaded surface generated by *surfl* function is then colored with the shades of single color. The results with increasing number of grid points viewed from different angles are shown below:

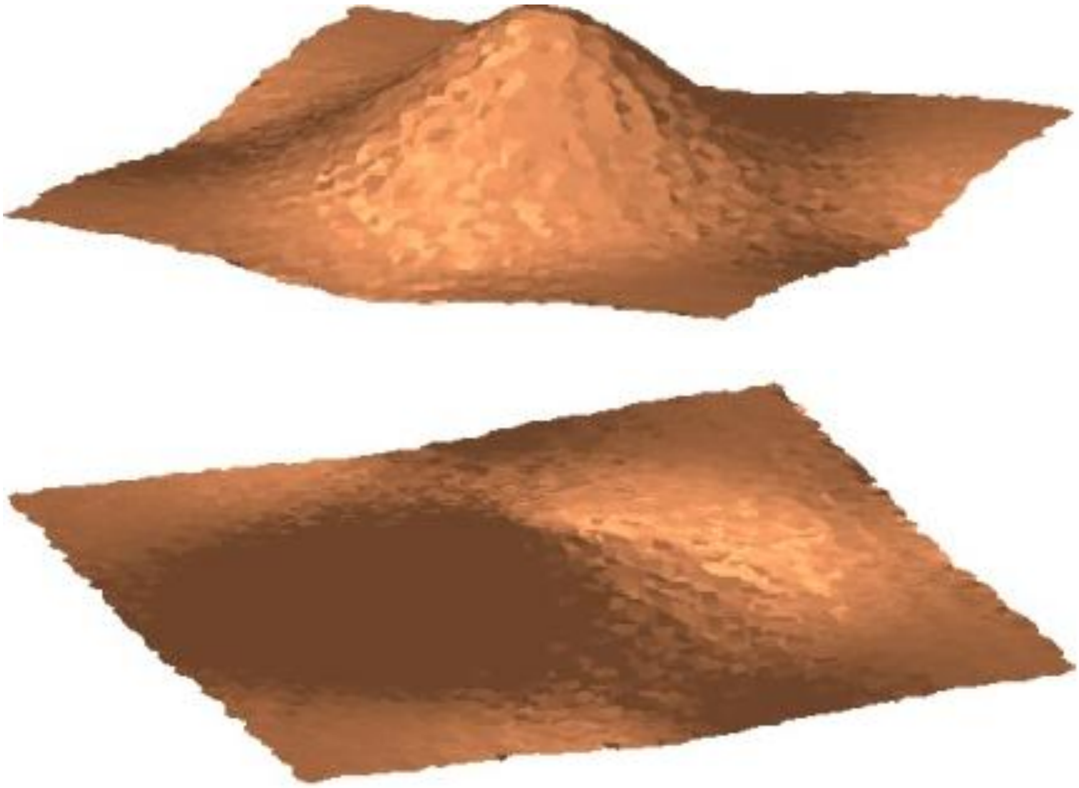


Figure-3 Single Hill Terrain viewed at different angles

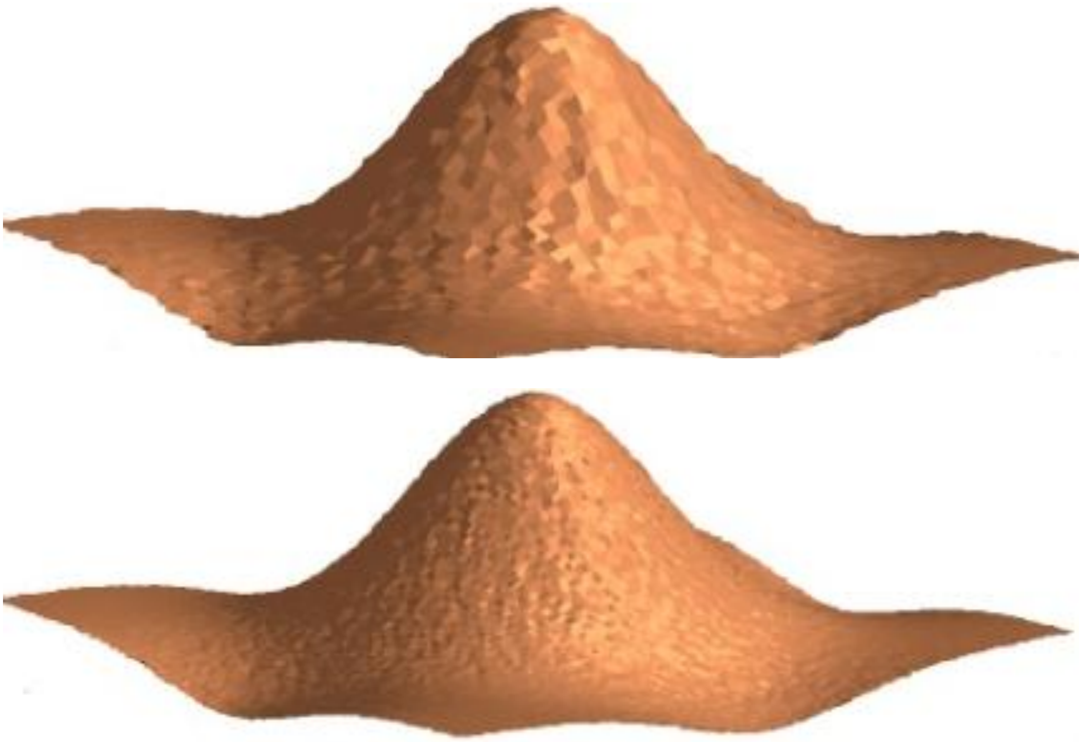


Figure-4 Single Hill Terrain Generated with Increasing Number of Grid Points

The images look more realistic in MATLAB figure window as it contains so many information visa the three coordinates, rendering and lighting values, camera positions, etc. These images can be artificially colored, rendered or edited using some readymade graphics tools or techniques for use in with some graphical or gaming applications.

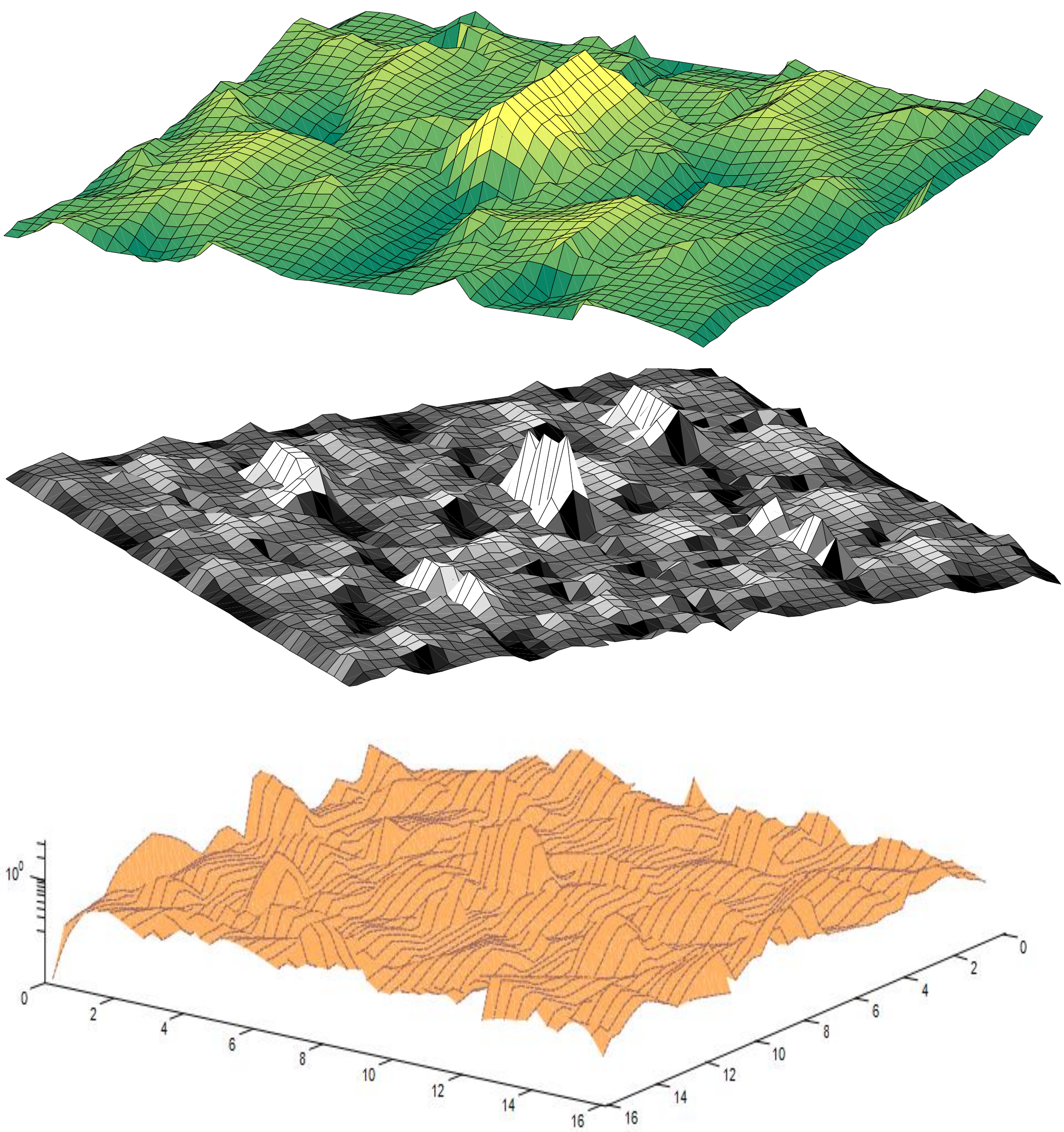


Figure-5 Multiple Hills Terrain shown with different colormaps

IX. CONCLUSION

Several complex algorithms are available to generate a terrain, but many of them do not provide any way to control the shape of the terrain. The terrain generation algorithm discussed here is much simple and the terrains are just created randomly. The terrains are generated with single hill and multiple hills. The shape of the terrains is controlled by z values in such a way that the top hill is cantered at the terrain grid (sub grid) midpoint(s) and as it goes towards sides the height of the terrain decreases. The terrain is generated in the form of (x,y,z), which is appropriate for many visualization applications.

X. SCOPE OF FUTURE WORK

The terrain generation has many applications in visualization. The basic idea behind a terrain landscape is to generate a continuous surface which varies in height randomly but with the random variation obeying a particular statistical law. The terrains generated here can be deformed further to add more realism.

XI. REFERENCES

[1] Donald Hearn & M. Pauline Baker, "*Random Midpoint Displacement Method"*, Computer Graphics, 2nd Edition, Pearson Education, page no. 373

[2] Miller, Gavin S. P., *"The Definition and Rendering of Terrain Maps"*. SIGGRAPH 1986 Conference Proceedings (Computer Graphics, Volume 20, Number 4, August 1986)

[3] Robot Frog, "*Hill Algorithm"*, http://www.stuffwithstuff.com/robot-frog/3d/hills/hill.html

[4] Chetan K. Sindhi, "*Single Hill Terrain Generation by Displacing Two Dimensional Grid Points*", International Journal of Advance Engineering and Research Development (IJAERD), Volume 02, Issue 03, March-2015