

Diamond-Square Algorithm Implementation Notes

The diamond-square algorithm was implemented iteratively and recursively. Both methods take as input two integers used to calculate the width and height of the field, a starting roughness value, a rate at which roughness decreases per iteration of the algorithm, and a scale.

With the two integer inputs w and h , the width and height of the field to be generated is calculated as $\text{width} = 2^w + 1$ and $\text{height} = 2^h + 1$.

The roughness value, called H in the code, is used to calculate the random number added to the average of the four points surrounding a center. Random numbers can be between $-H$ and H .

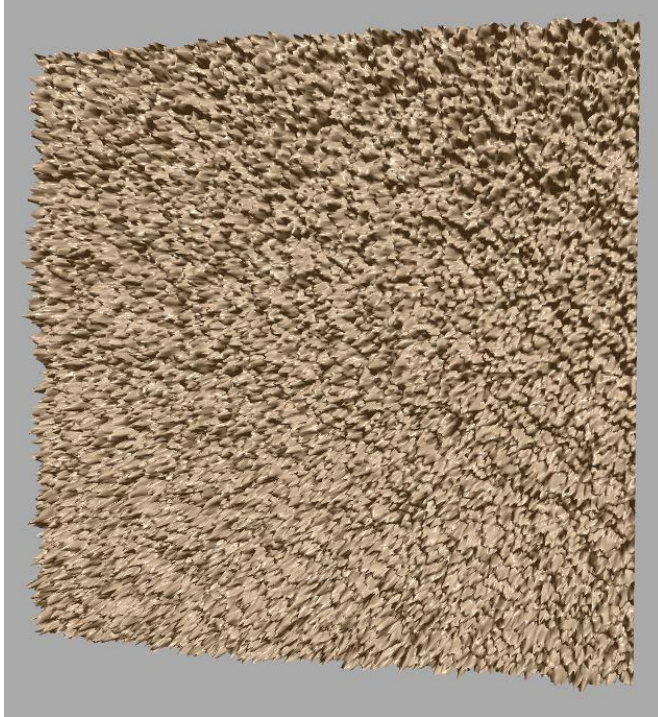
The parameter that determines the rate at which H decreases is called f . This is a value between 0 and 1. After each iteration, the new value of H is calculated as $H = H / 2^f$.

Finally, the scale increases the size of the plane geometry that displays the height field created by the diamond-square algorithm at the end of the process.

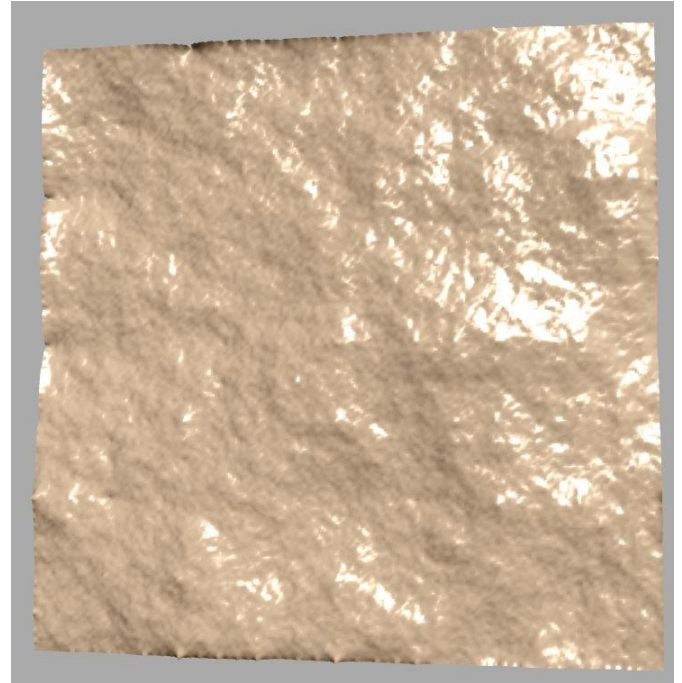
Results

Iterative diamond-square results

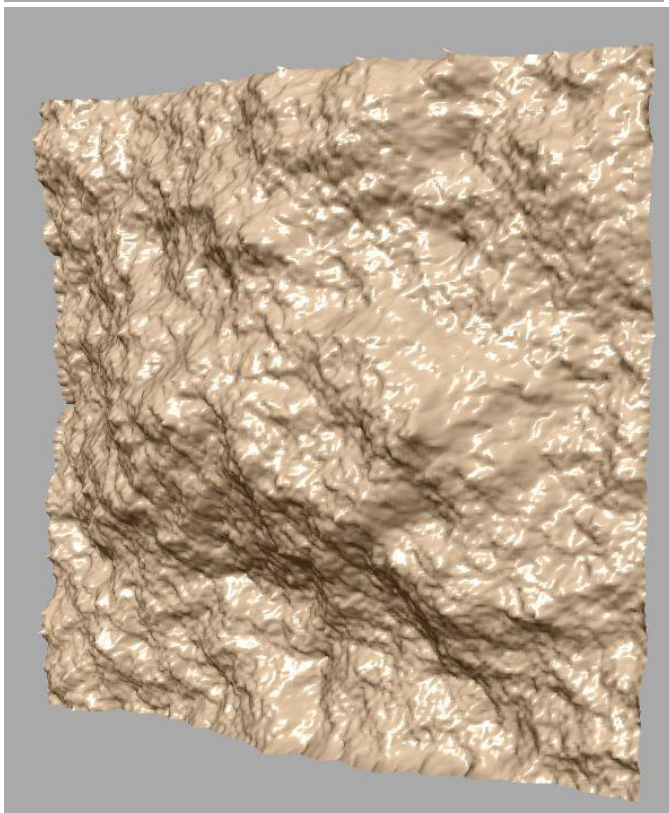
$H = 20, f = 0$



$H = 50, f = 1$

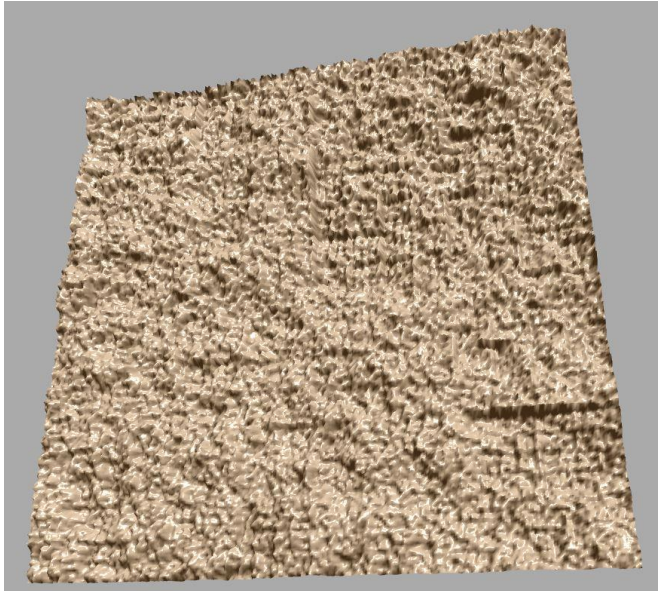


$H = 200, f = 1$

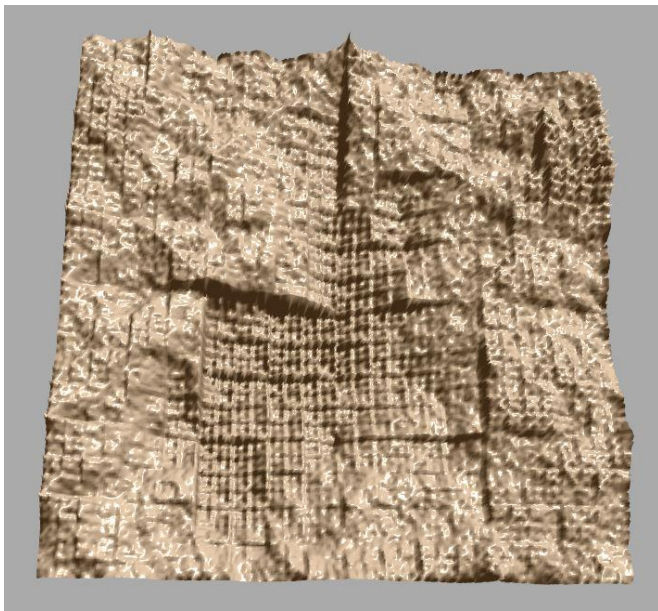
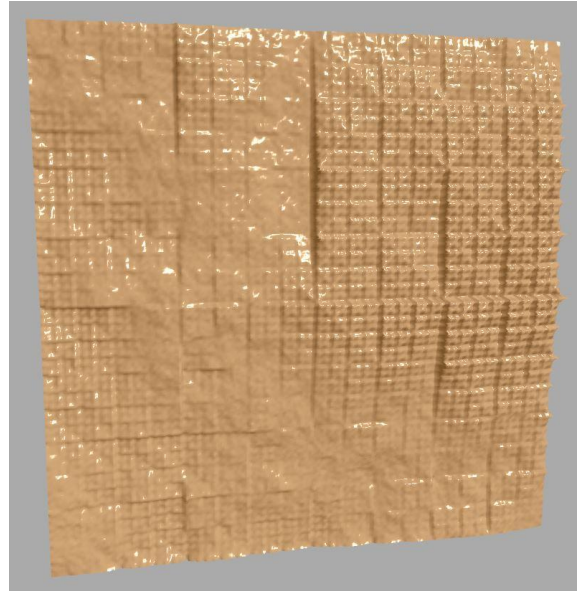


Recursive diamond-square results

$H = 50, f = 0.5$



$H = 50, f = 1$



$H = 200, f = 1$

The squares used to generate the surface were more prominent when the recursive diamond-square algorithm was used. It's likely this is because the height of some of the points outside the current square being processed were undefined when the height of the centerpoints in the square step were calculated. The squares became less prominent the smaller the value of f , the more the height is allowed to vary between recursive steps.

Height field from image results

