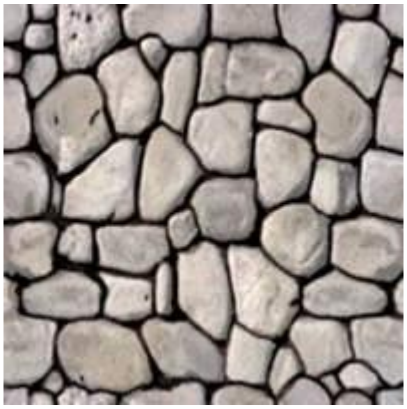
CS116 hw3

Jenny Zeng

52082740

zhaohuaz@uci.edu

Test on rock_wall.jpg

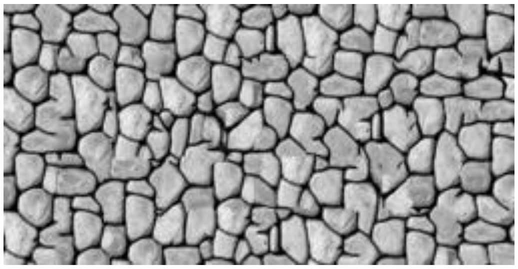


default setting:

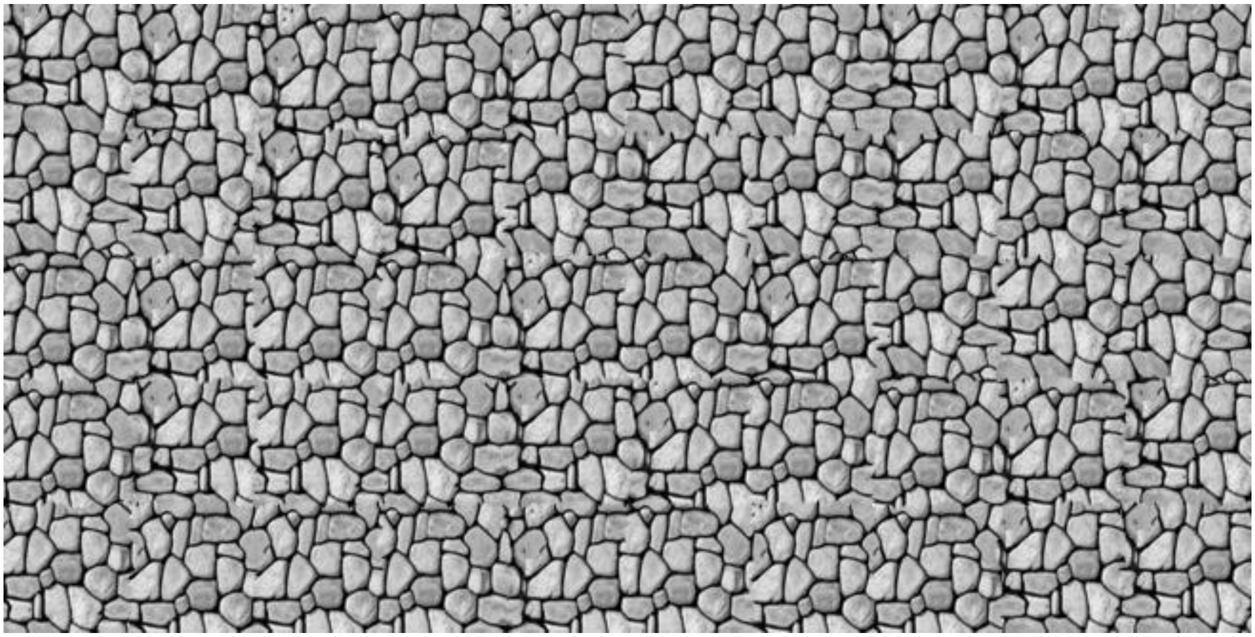
tilesize = 30;

overlap = 5;

K = 5;

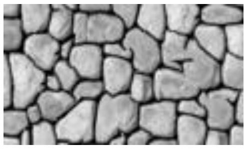


(1) *increase the tile size*
tilesize = 70;
overlap = 5;
K = 5;



As tile size increases, the size of the output image increases and we can find many places are very similar with each other. Looking into the sampletiles.m file, we can find that the function extracts all tilesize by tile size sub regions of an image. If the tile size is increased, it will extract a larger sub region of an image. When tilesize = 70, duplicate regions in the figure become very obviously because the sub region is so big that it almost covers the whole image. Since the size of the sample tiles is increased due to this reason, the output figure size also increases.

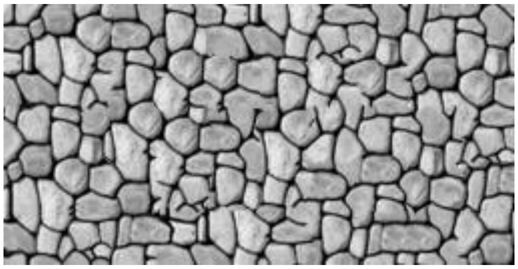
(2) *increase overlap*
tilesize = 30;
overlap = 30;
K = 5;



The size of the output figure is smaller. This is because many places are overlapped and then the total size would be smaller because of the formula for the output figure size in the stitch function (width = W1+W2 – overlap).

It also has a zoom-in effect because the non-overlap region between each image becomes smaller.

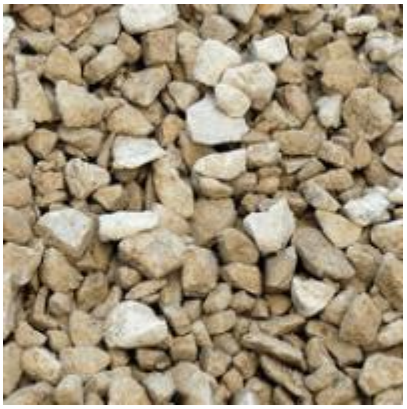
(3) *decrease the value for K*
tilesize = 30;
overlap = 5;
K = 1;

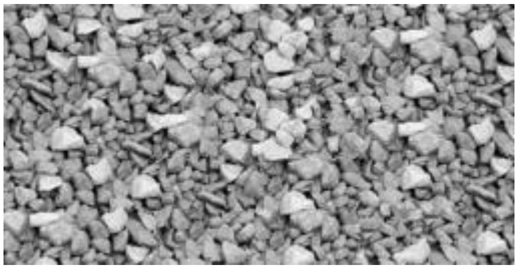


It is hard to find any changes when the original figure and the one with a smaller K value. Through looking into the code, we can see that it is finding a match from its top K nearest neighbors. When the number of sample tiles is large, it does not hamper it from finding a good match no matter k is large or small.

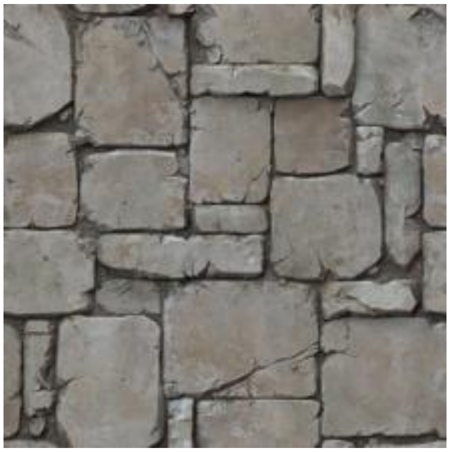
two other texture images

1. input





2. input:



output: