ME 557 HW 4

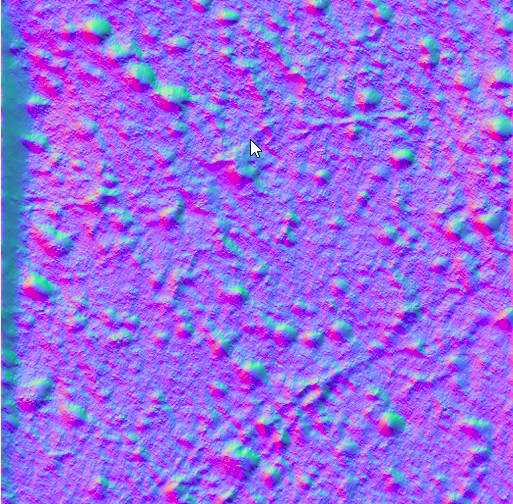
Yuanfen Chen, Michael Hoefer, Marcus Naik

Problem 2: Using noisemaps

We started with this original image of the chicago skyline, which is our “landscape.”



Next, we created a noisemap from the image of a sponge, shown below.



noiseVec = normalize(texture(noiseMap, pass\_TexCoord).xy);

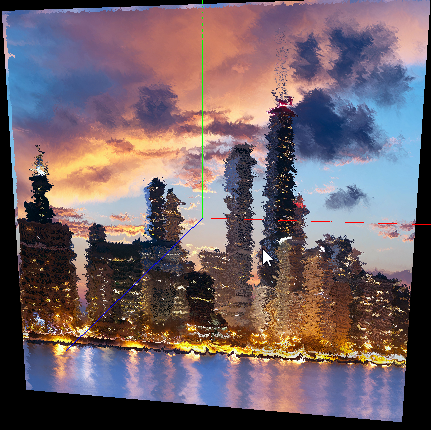
The xy could be changed to yz, xz, corresponding to RG, GB, RB color value. After normalization, the noiseVec is a vector of 2 with value within 0-1.

noiseVec = (noiseVec \* 2.0 - 1.0) \* 0.035;

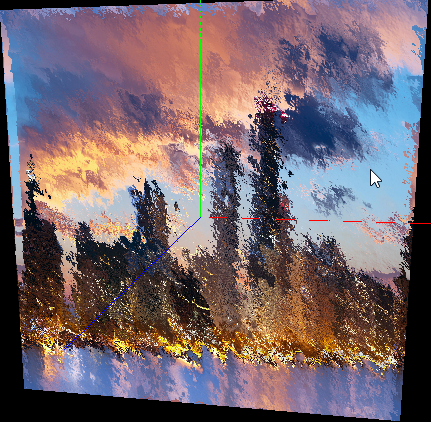
(noiseVec \* 2.0 - 1.0) change the value range of the noiseVec, now the noiseVec could be negative, which makes it possible for the landscape to shift to every direction. (+,+) noiseVec shift the landscape to downleft, (+,-) noiseVec shift the landscape upleft, noiseVec(-,-) shift the landscape upright, noiseVec(-,+)shift the landscape downright.

The 0.035 is the scaling factor, which scale down the magnitude of the noise.

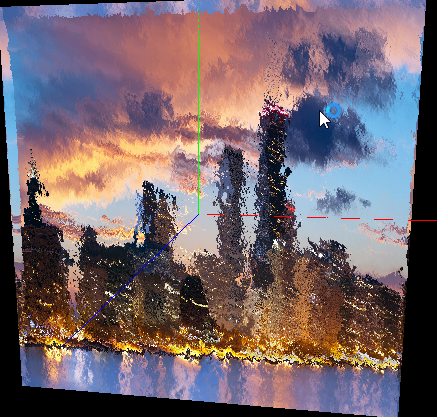
Using the equation given: noiseVec = (noiseVec \* 2.0 - 1.0) \* 0.035; and the YZ (gb) values of the noisemap resulted in the following image. The left edge now is actually the right edge in the original figure; the upper edge is actually the down edge in the original. That is because after (noiseVec \* 2.0 - 1.0), the noiseVec is negative at the noise map, so the pass\_TexCoord + noiseVec became negative, say, -0.05, then it seems the computer add 1 to the negative value, ending in a value close to 1, say 0.095. Same for the upper edge.



By using the xy values instead yz, the following image is displayed.Using the xy (rg) components of the noisemap makes the noisiest image. This is likely because the red and blue colors are most prominent (higher values) in the noisemap image, so the distortion is higher.

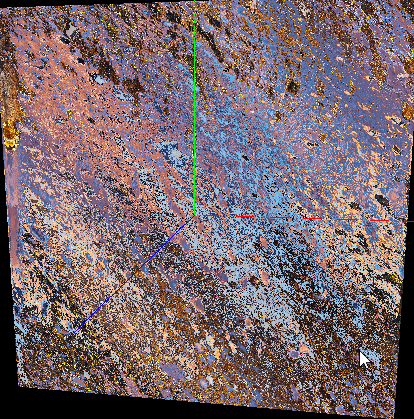


Same for xz(rb),the noisemap makes the noisiest image

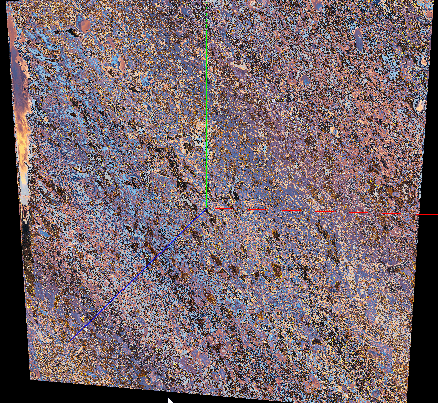


Sticking with the xy components (rg), we altered the blending equation by increase the scaling factor. This makes the noiseVec larger and results in bigger distortion, since a larger value is added to the original texel.

noiseVec = (noiseVec \* 2.0 - 1.0) \* 0.5;

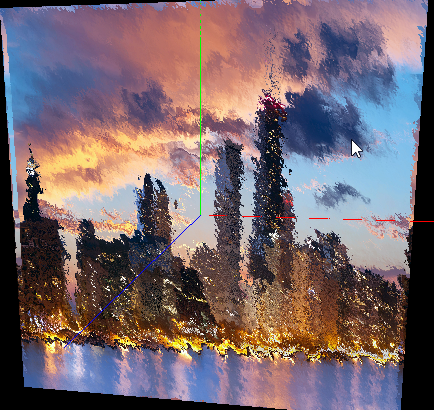


Next, changing the scaling factor to 1 resulted in an even noisier picture.



Next, we decreased the scaling factor below its original value of .035

noiseVec = (noiseVec \* 2.0 - 1.0) \* 0.02;



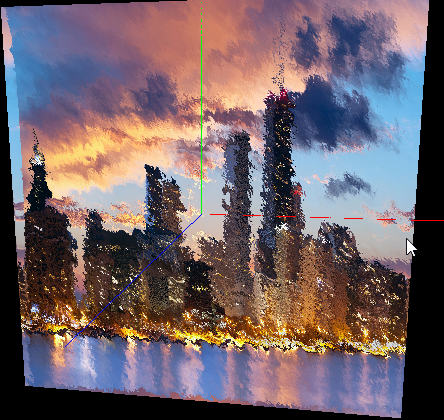
Changing the scaling factor to .005 resulted in an almost clear picture. One thing to note is that as the distortion decreases, the “weird space” along the edges of the picture gets less and less.



Next, we experimented with the other variables.

We reduced the multiplication from 2 to 1.

noiseVec = (noiseVec \* 1.0 - 1.0) \* 0.035;



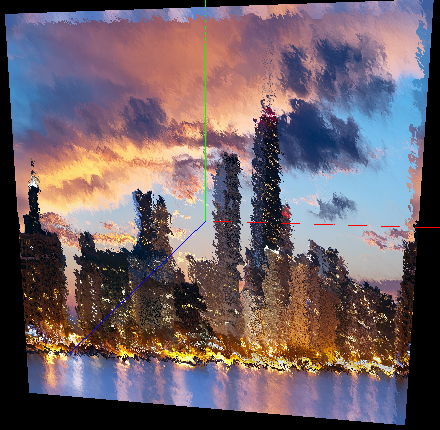
Changing the variable to 5 results in more noise.

noiseVec = (noiseVec \* 5.0 - 1.0) \* 0.035;



When we remove the \*2 -1 part of the equation, the “weird part” of the image shifts to the other side. Part of the building on the left side of the original image is now on the right side of the altered image. The sky also has some elements of the water.

noiseVec = (noiseVec ) \* 0.035;



It seems that most of the variables change the distortion of the image. This makes sense, because it alters which pixel of the texture is mapped onto the primitive.