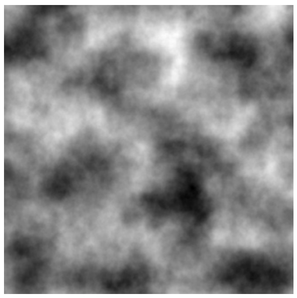
Comparison of Procedural Noise Generation through Varying Pairing Functions

This comparison was based on the amplitude histogram and Fourier transform of each noise image generated under five different pairing functions. The five pairing functions used were Linear Pairing, Cantor Pairing, Szudzik Pairing, Rosenberg-Strong Pairing, Original Perlin Noise Pairing.

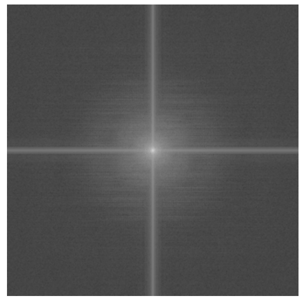
The analysis was performed with and without the use of a permutation table, in which the use case for the pairing functions differed slightly. With the use of a permutation table, the hash functions are used to determine an index of the permutation table (which is then used as an index to the gradient arrays). Without the use of a permutation table, the hash functions are used to determine an index of the gradients arrays. These can be seen in the Appendix and are also located in PDF for in the repository.

Analysis with Permutation Table

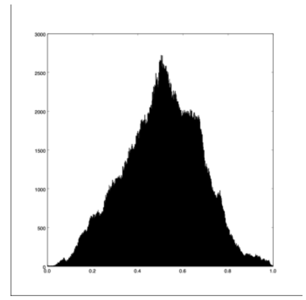
The original Perlin noise outputs, using a permutation table, will be used as a baseline for comparison. This is as follows:



Noise



Fourier



Amplitude

We can observe that the Fourier transform has subtle horizontal lines occurring around the center of the output which is indicative of some sort of correlation when the original hash function proposed by Perlin is used. Being able to remove such a noise could prove useful in reaching more true random noise. Per the paper “A Survey of Procedural Noise Functions”, the following is stated with regard to the horizontal lines occurring:

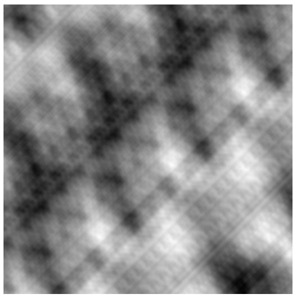
> *“The horizontal features in the periodogram of Perlin noise are caused by an undesired correlation in the hash function.”*

<u>With the use of a permutation table</u>, it can be noted that all outputs exhibit similar patterns in their noise output, amplitude histogram and Fourier transform. However, there are slight variances in the appearance seen at the center of the Fourier transfer.

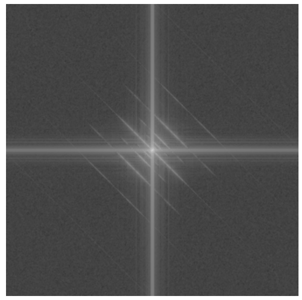
	Fourier
Linear	Appearance of vertical lines appearing close to center. Also indicative of correlation in the hash function.
Cantor	Appearance of diagonal lines appearing close to center. Also indicative of correlation in the hash function. Similar appearance to the original Perlin hash function but off axis by 45 degrees.
Szudzik	No visible appearance of lines appearance, indicating less correlation that previously seen in the Original Perlin hash function, along with the Linear and Cantor pairing functions.
Rosenberg-Strong	Very similar output to the Szudzik pairing function, but emits three subtle diagonal lines (significantly less lines than Linear, Cantor or the Original Perlin hashing functions).

Analysis without Permutation Table

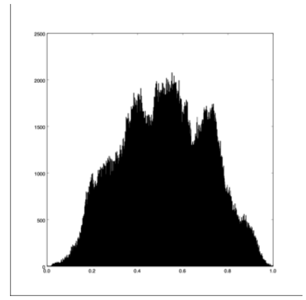
The original Perlin noise outputs, without the use of a permutation table, will be used as a baseline for comparison. This is as follows:



Noise



Fourier



Amplitude

We can observe that the Fourier transform has diagonal lines occurring around the center of the output which is indicative of some sort of correlation when the original hash function proposed by Perlin is used. Evidently, the noise generated does not exhibit the randomness desired. Therefore, the original Perlin noise hash function is certainly reliant on the permutation table existing.

Without the use of a permutation table, it can be noted that the noise and Fourier transform differ with the use of the hash functions that aren't the original Perlin hash function. Its clear how the noise output differs, per the above image. The Fourier analysis differs greatly from the other four pairing functions in the following regard:

	Fourier
Linear	Appearance of vertical lines appearing close to center. Also indicative of correlation in the hash function.
Cantor	Appearance of diagonal lines appearing close to center. Also indicative of correlation in the hash function. Similar appearance to the original Perlin hash function, but more subtle in the way the lines appear.
Szudzik	No visible appearance of lines appearance, indicating less correlation that previously seen in the Original Perlin hash function, along with the Linear and Cantor pairing functions.
Rosenberg-Strong	Very similar output to the Szudzik pairing function, but emits three subtle diagonal lines (significantly less lines than Linear, Cantor or the Original Perlin hashing functions).

Conclusion

The following can be drawn from the above analysis, with respect to each pairing function:

Linear

There is clearly correlation between the pairing function and the output (i.e. lines appear in Fourier), both with and without the use of a permutation table, as is witnessed in the original Perlin noise pairing function. This is undesirable as noted by Ken Perlin in the paper “A Survey of Procedural Noise Generation”.

Cantor

Both with and without the permutation table, this output emits an extremely similar to output to that of the original Perlin noise pairing function (with a permutation table). There is clearly correlation between the pairing function and the output (i.e. lines appear in Fourier), both with and without the use of a permutation table, as is witnessed in the original Perlin noise pairing function. This is undesirable as noted by Ken Perlin in the paper “A Survey of Procedural Noise Generation”.

This method generates an output that most closely models that of the Perlin noise hashing function. As such, it could be proposed that this hashing function could be used to replace the need for a permutation table.

Szudzik

This output was likely the most intriguing of the pairing functions being compared. Both with and without the permutation table, this output emits an extremely similar to output to that of the original Perlin noise pairing function (with a permutation table), but with the lack of any lines appearing.

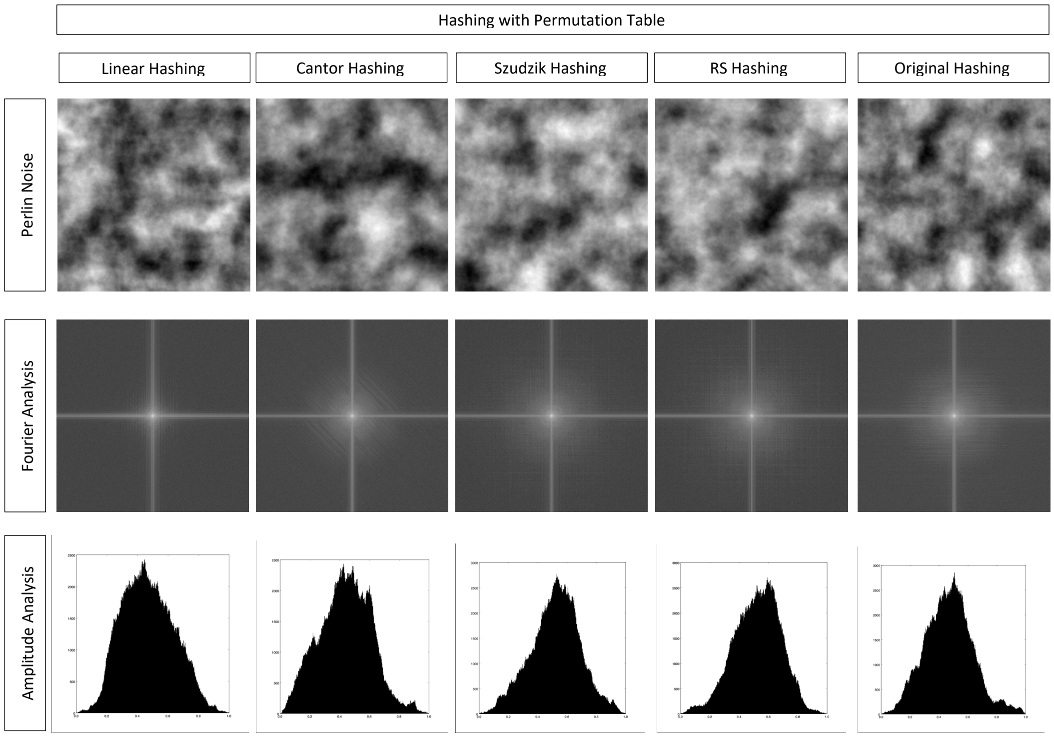
As such, there is an argument to be made that this pairing function results in less correlated result in the noise generated, thus enhancing the randomness of the overall noise generated. Much alike the Cantor hashing function, this could also replace the need for a permutation table as the same results are yielded both with and without.

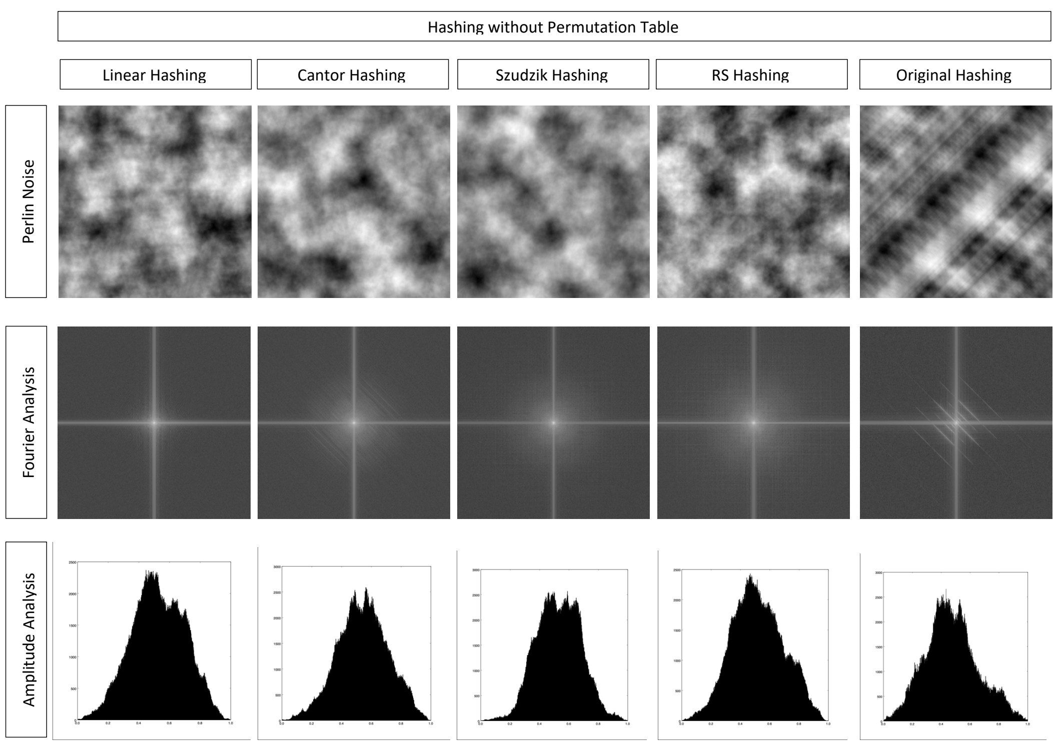
Rosenberg-Strong

Both with and without the permutation table, this output emits an extremely similar to output to that of the original Perlin noise pairing function (with a permutation table), but with the a major reduction in the number of lines that appear.

As such, there is an argument to be made that this pairing function results in less correlated result in the noise generated, thus enhancing the randomness of the overall noise generated. Much alike the Cantor hashing function, this could also replace the need for a permutation table as the same results are yielded both with and without.

Appendix


Hashing with Permutation Table
Linear Hashing
Cantor Hashing
Szudzik Hashing
RS Hashing
Original Hashing
Perlin Noise
Fourier Analysis
Amplitude Analysis


Hashing without Permutation Table
Linear Hashing
Cantor Hashing
Szudzik Hashing
RS Hashing
Original Hashing
Perlin Noise
Fourier Analysis
Amplitude Analysis