Karlsruhe Institute of Technology

Zeitlich stabile blue noise Fehlerverteilung im Bildraum für Echtzeitanwendungen

Bachelorarbeit von

Jonas Heinle

An der Fakultät für Informatik
Institut für Visualisierung und Datenanalyse,
Lehrstuhl für Computergrafik

16. November 2019

Inhaltsverzeichnis

1. Prelude

1.1 Abstract

1.2 Einleitung

2. Grundlagen

2.1 Path Tracer

2.1.0.1 Funktionsweise

...

2.2 Blue Noise

2.2.1 Eigenschaften

Wie in [Gam17] vorgestellt, macht man sich die Eigenschaften einer blue noise Textur zu Nutze.

2.2.1.1 Uniformität

Die Uniformität(spätlat. *uniformitas*-Einförmigkeit)

2.2.1.2 Niedrige Frequenzen

Niedrige Frequenzen sind in einer blue noise sehr wenig bis gar nicht vertreten.

2.2.1.3 Isotropie

Die Isotropie(altgr. *isos*-gleich und *tropos*-Richtung) einer blue noise Textur wird ausgenutzt. Dabei haben wir in allen Dimensionen (in dieser Arbeit werden Texturen mit zwei benutzt) die Unabhängigkeit einer Eigenschaft.

2.2.1.4 Kachelung

Eine weitere nützliche Eigenschaft der blue noise Verteilung ist die Möglichkeit der Kachelung.

...

3. Temporaler Algorithmus

...

3.1 Sorting

(replace dummy code with correct code) **ToDo**

Algorithm 1 Sortier Schritt t nach dem Rendern von Frame t und vor dem Rendern von Frame t+1

```
1: pixel consists of value,index;
2: List framePixelsIntensities, noiseIntensities;
3: List L ← pixels in block
4: //init lists
5: for all (i,j) ← L do
6:     framePixelsIntensities(i,j) = pixelIntensity(frame(i,j));
7:     noiseIntensities(i,j) = pixelIntensity(blueNoise(i,j));
8: end for

9: //sort the two lists by means of intensities
10: sort(framePixelsIntensities);
11: Sort(noiseIntensities);

12: //now we reorder our seeds hence the sorted lists
13: for all i = 1..numberOfPixelsPerBlock do
14:     sortedSeeds(noiseIntensities.getIndex(i)) = incomingSeeds(framePixelIntensities.getIndex(i));
15: end for
```

...

3.2 Retargeting

(replace dummy code with correct code) **ToDo**

...

Algorithm 2 Retargeting Schritt t Vor Rendern Frame t+1 nach Sortier Schritt

```
1: //permutation indices from precomputed texture
2: List<PixelPermutation> L = getRetargetedSeedsIndices(incomingSeeds);
3: for all i = 1 .. numberOfPixelsPerBlock do
4:    retargetedSeeds(L.getPermIndices()) = incomingSeeds(L.getOldIndices());
5: end for
```

Literaturverzeichnis

[Gam17] Epic Games: *The problem with 3d blue noise*, 2017. `http://momentsingraphics.de/3DBlueNoise.html`, `Blogpost`.

Erklärung

Ich versichere, dass ich die Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet habe. Die Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt und von dieser als Teil einer Prüfungsleistung angenommen.

Karlsruhe, den 16. November 2019

(Jonas Heinle)