GPU Random Numbers via the Tiny Encryption Algorithm

Fahad Zafar, Marc Olano and Aaron Curtis

University of Maryland Baltimore County

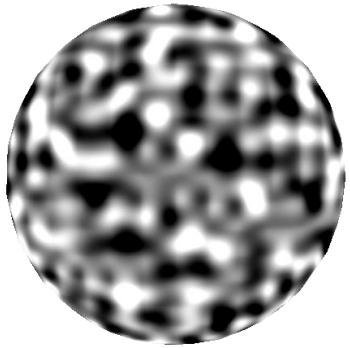
Outline

- Introduction
- Previous Work
- Analysis and Results
- Monte Carlo Shadow Algorithm
- Conclusion

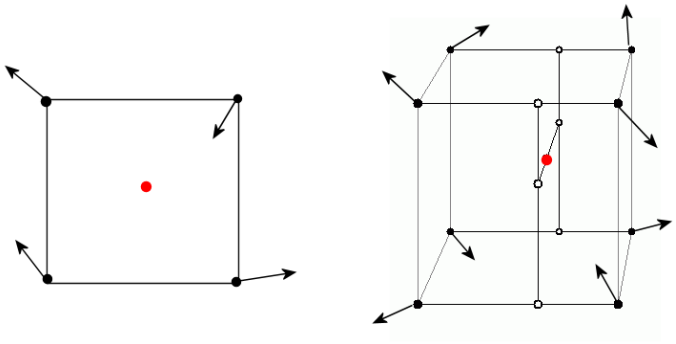
Introduction

- GPUs need random numbers
  - Real time graphics, Simulations, GPGPU
- Required characteristics
  - Repeatability
  - Random access
  - Multiple independent streams
  - Speed
  - Minimal Statistical bias

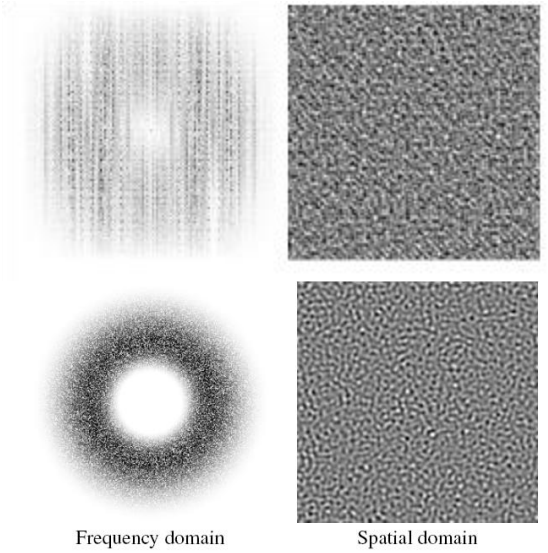
Gradient Noise



- Use pseudo random gradient vectors at each lattice point and use them to generate the noise value.

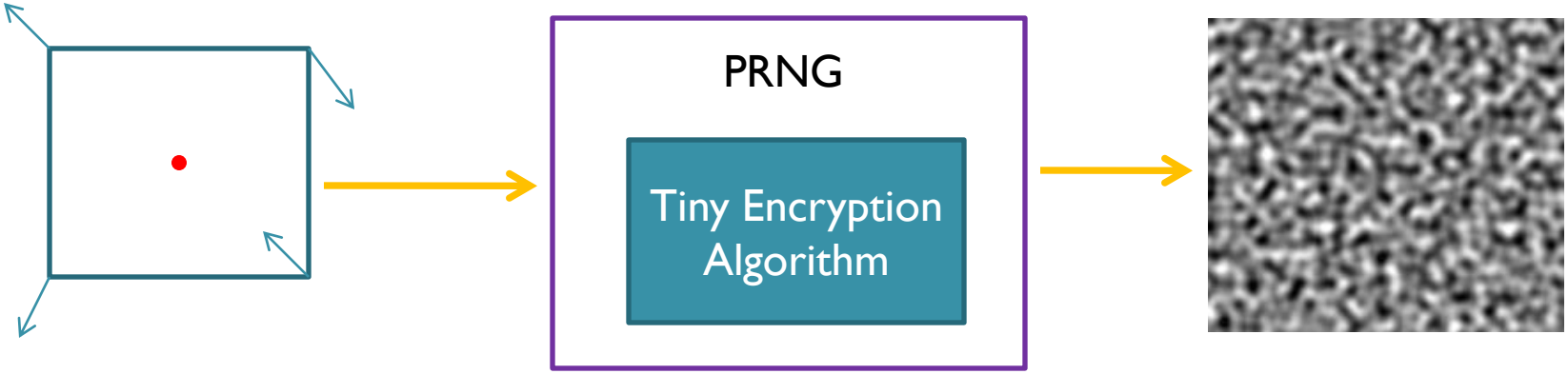


Visualizing 2D and 3D gradient noise neighboring vectors

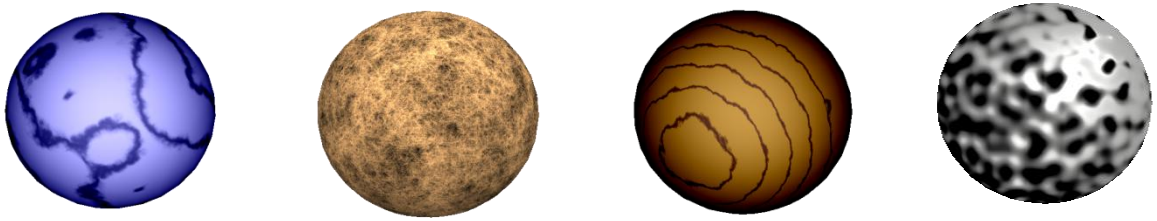


(*Left*) 2D Fourier Transform
(Right) 2D Noise Tile

Tiny Encryption Algorithm for gradient noise



2D/3D/…..

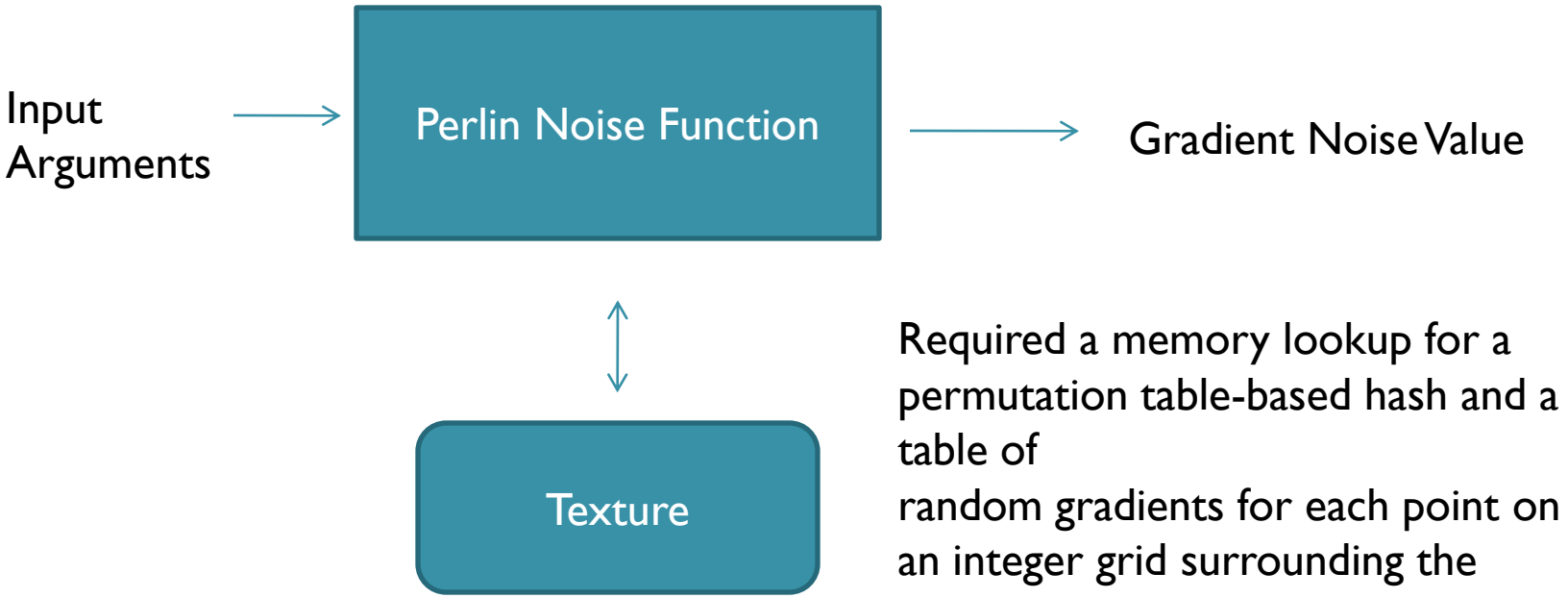


Outline

- Introduction
- Previous Work
- Analysis and Results
- Monte Carlo Shadow Algorithm
- Conclusion

Previous Work

- The Image Synthesizer Perlin [1985] and Improving Noise papers by Perlin [2002]



Previous Work

Wavelet Noise is an alternative to Perlin noise which reduces the problems of aliasing and detail loss, that are encountered when Perlin noise is summed into a fractal.

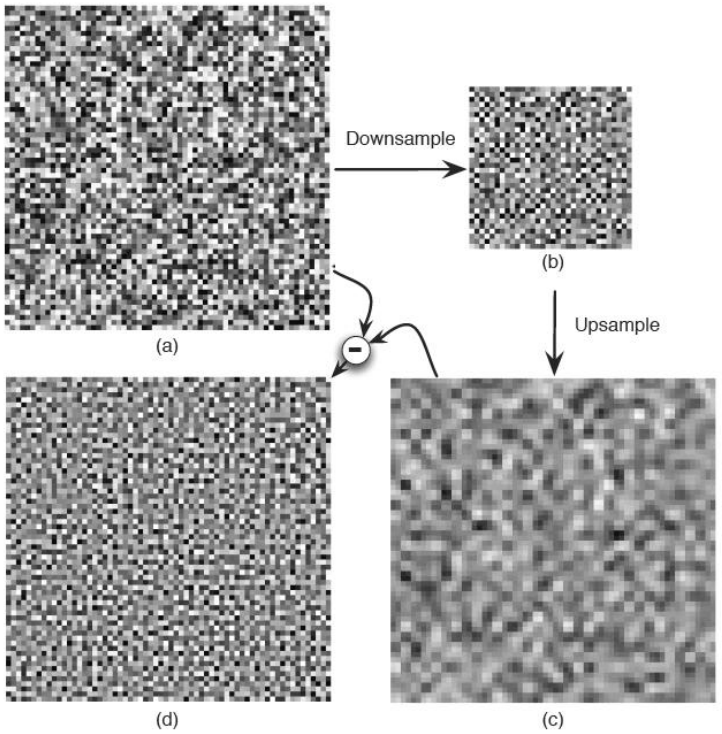
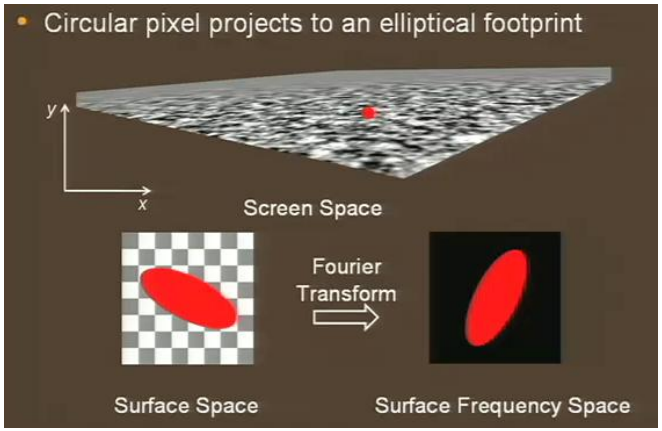
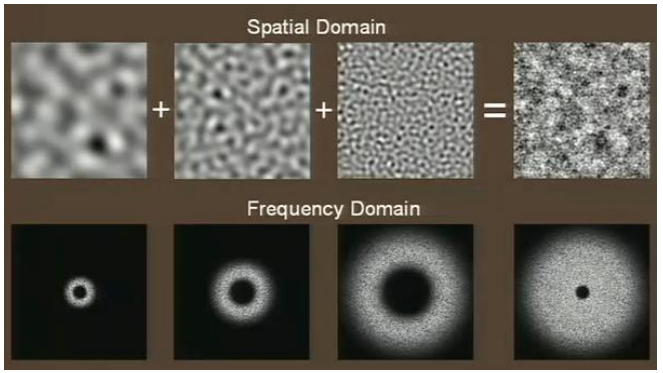


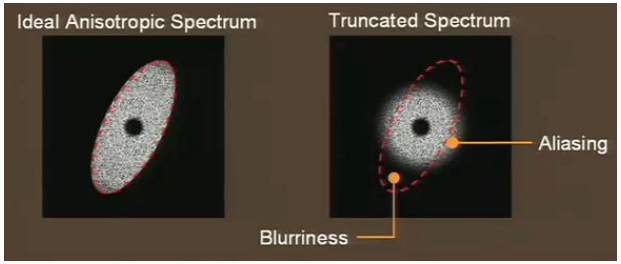
Figure 2: *(a). Image R of random noise, (b) Half-size image* $R^{\downarrow}$*, (c) Half-resolution image* $R^{\downarrow\uparrow}$*, (d) Noise band image* $N = R - R^{\downarrow\uparrow}$.

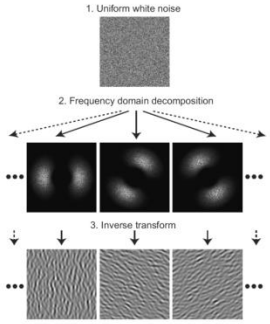
Image Reference : Wavelet Noise Cook et al. 2005

- Anisotropic Noise by Goldberg et al. [2008]
  - Blend of directional noise stored in different channels of a texture to reduce detail loss







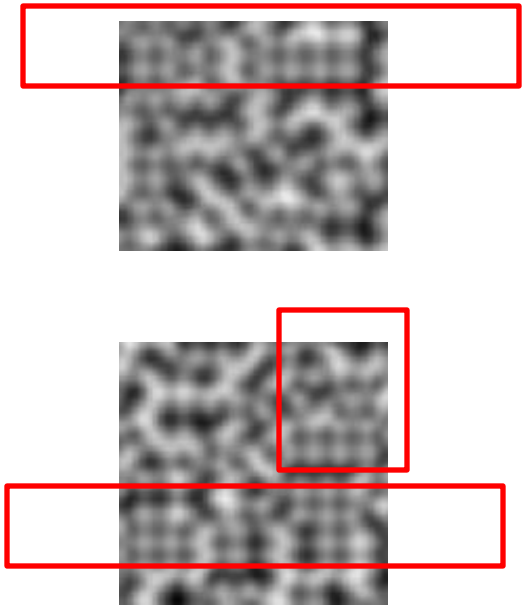


Images Reference : Anisotropic Noise Goldberg et al. 2008

Modified Noise by Olano 2005

- Used Blum Blum Shub to create permutations on the fly.
- No texture access
- Very Fast and Simple
- Bad quality output

$$Hash(x) = X^2 \text{ MOD } M,\ M = 61$$



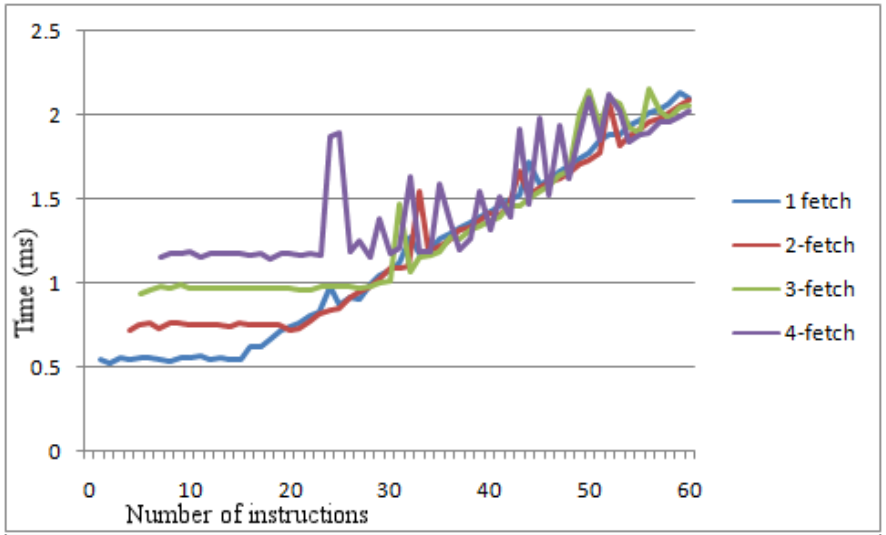
- White Noise Generation via Cryptographic Hash (Tzeng and Wei [2008])
  - Use MD5 instead of Blum Blum Shub.
  - Quality random numbers.
  - Very Slow(64 rounds)
    - (adds to number of instructions)
  - 4 Types of rounds

Algorithm	DH Tests Passed
MD5GPU	15 / 15
GPU CEICG	10 / 15
GPU BBS	2 / 15
GPU AES	7 / 15
Goulburn	9 / 15
`rand`	6 / 15
`drand48`	12 / 15
M. Twister	15 / 15

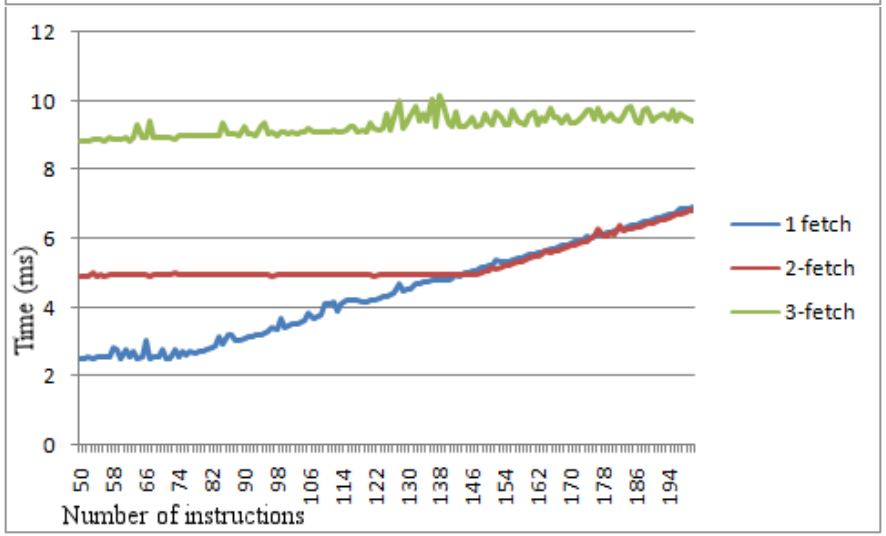
Outline

- Introduction
- Previous Work
- Analysis and Results
- Monte Carlo Shadow Algorithm
- Conclusion

Compute bound shader analysis



Sequential Texture Access



Random Texture Access

GPU: NVIDIA 8600M

Searching for a Faster Hash

- RC4; which requires a lookup table [Golic 1997]
  - Stream cipher
  - Key scheduling algorithm (initialize)
- CAST-128; which requires a large amount of data for initialization
  - is a block cipher
  - There are three alternating types of round function.
- Blowfish; which was fast but each new key required pre-processing equivalent to encrypting nearly 4 kilobytes of data [Schenier 1996]

Tiny Encryption Algorithm (TEA)

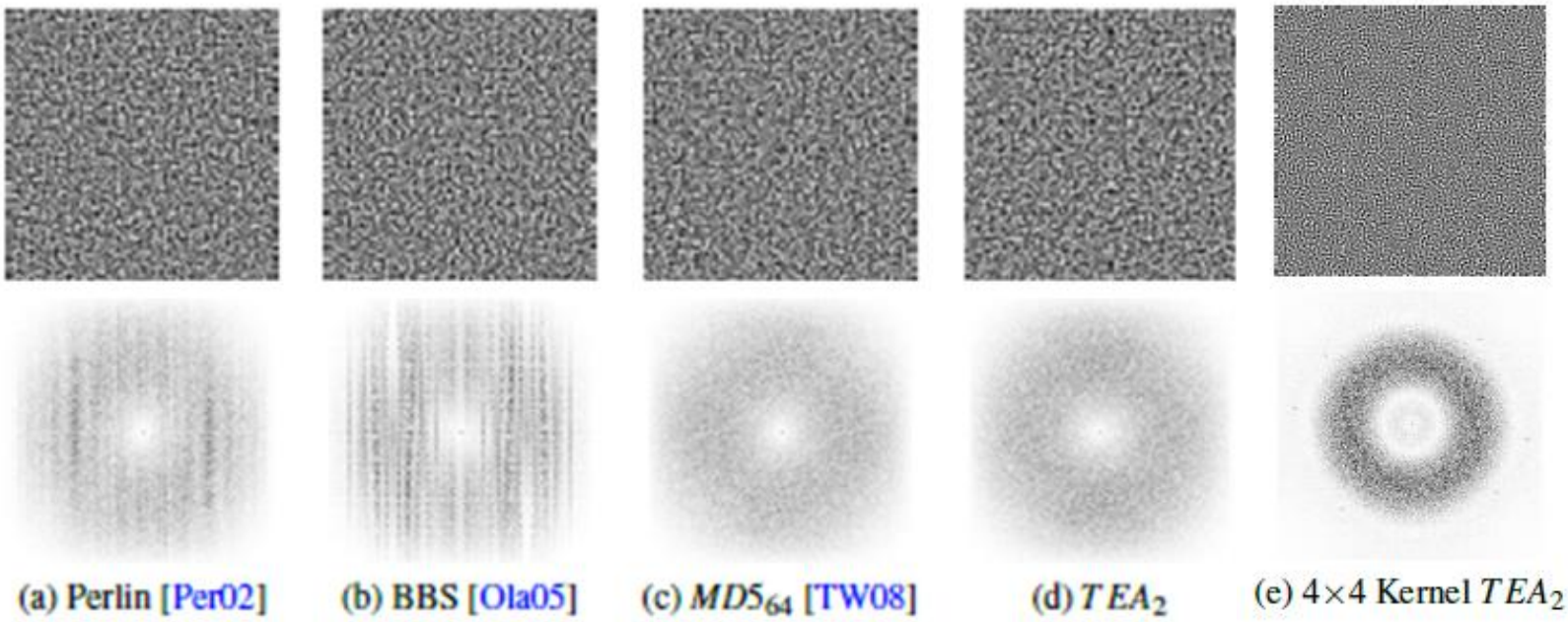
- Used extensively on legacy computers
- Analyzed by Reddy [2003] (Defeating TEA)
- Perfect match to our requirements
- XTEA and XXTEA [Needham et al. 1997]

One Round

```
sum += 0x9e3779b9;
v0 += ((v1 << 4) + 0xA341316C)^(v1 + sum) ^ ((v1 >> 5)+ 0xC8013EA4);
v1 += ((v0 << 4) + 0xAD90777D)^(v0 + sum) ^ ((v0 >> 5)+ 0x7E95761E);
```

XOR	SHIFTS	ADDITIONS
4	4	9

Gradient Noise Fourier Transforms



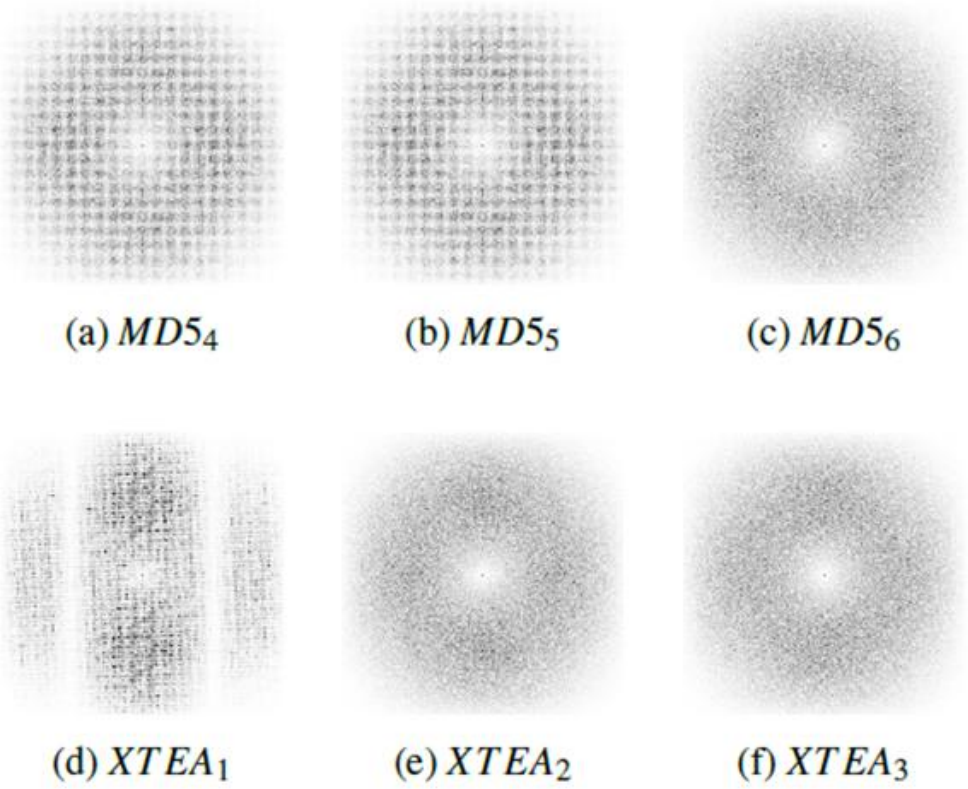
(a) Perlin [Per02] (b) BBS [Ola05] (c) $MD5_{64}$ [TW08] (d) TEA_2 (e) 4×4 Kernel TEA_2

Timing Noise with different PRNGs in Mpixels/sec

	2-D Noise		*4 × 4 Kernel*	
RNG	**HD 4870**	**8600M**	**HD 4870**	**8600M**
BBS	3077	1628		
$MD5_{64}$	173	61	27	10
TEA_8	1000	322	281	81
TEA_2	3243	1305	1000	364
$XTEA_{16}$	635	184	187	40
$XTEA_2$	3529	1551	1143	419

Per round frequency domain analysis for noise

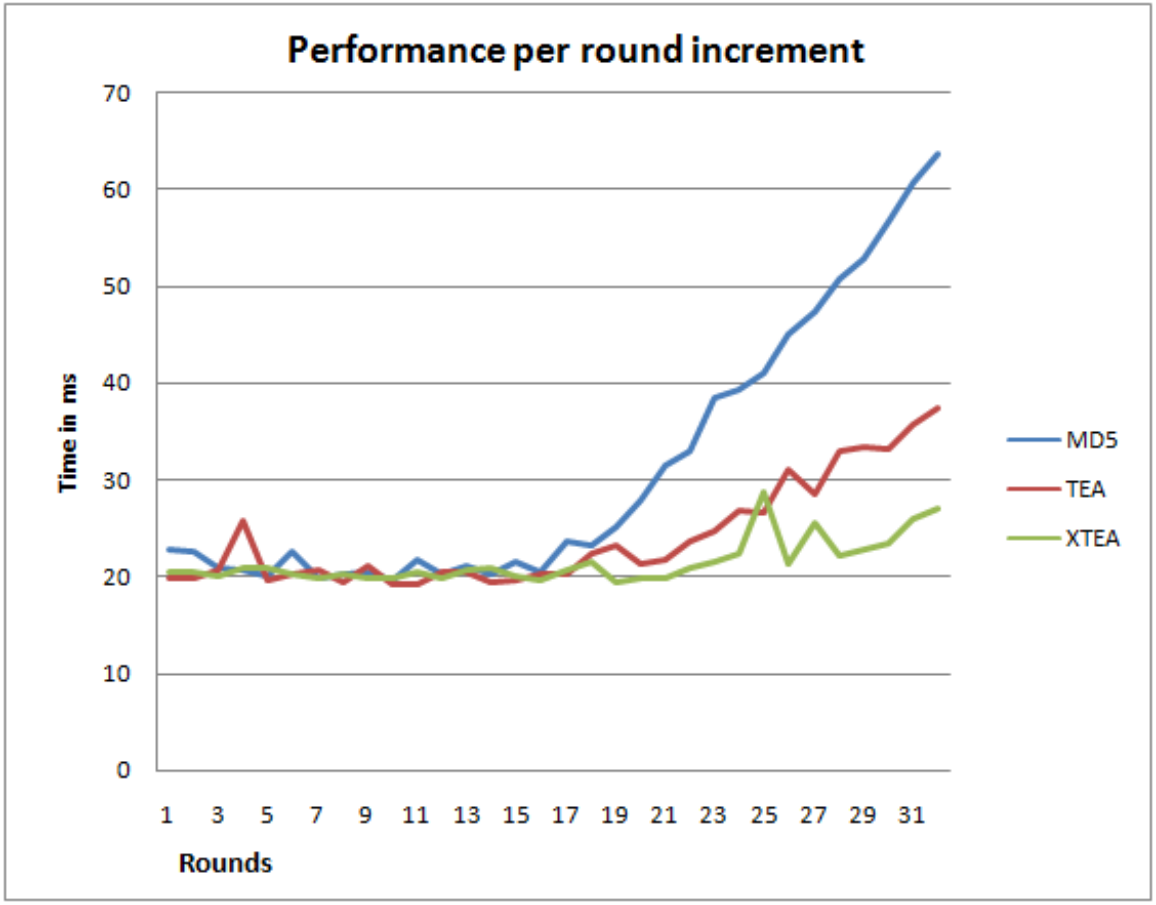
- Good quality noise can still be produced with a lower quality random number generator.



(a) $MD5_4$ (b) $MD5_5$ (c) $MD5_6$

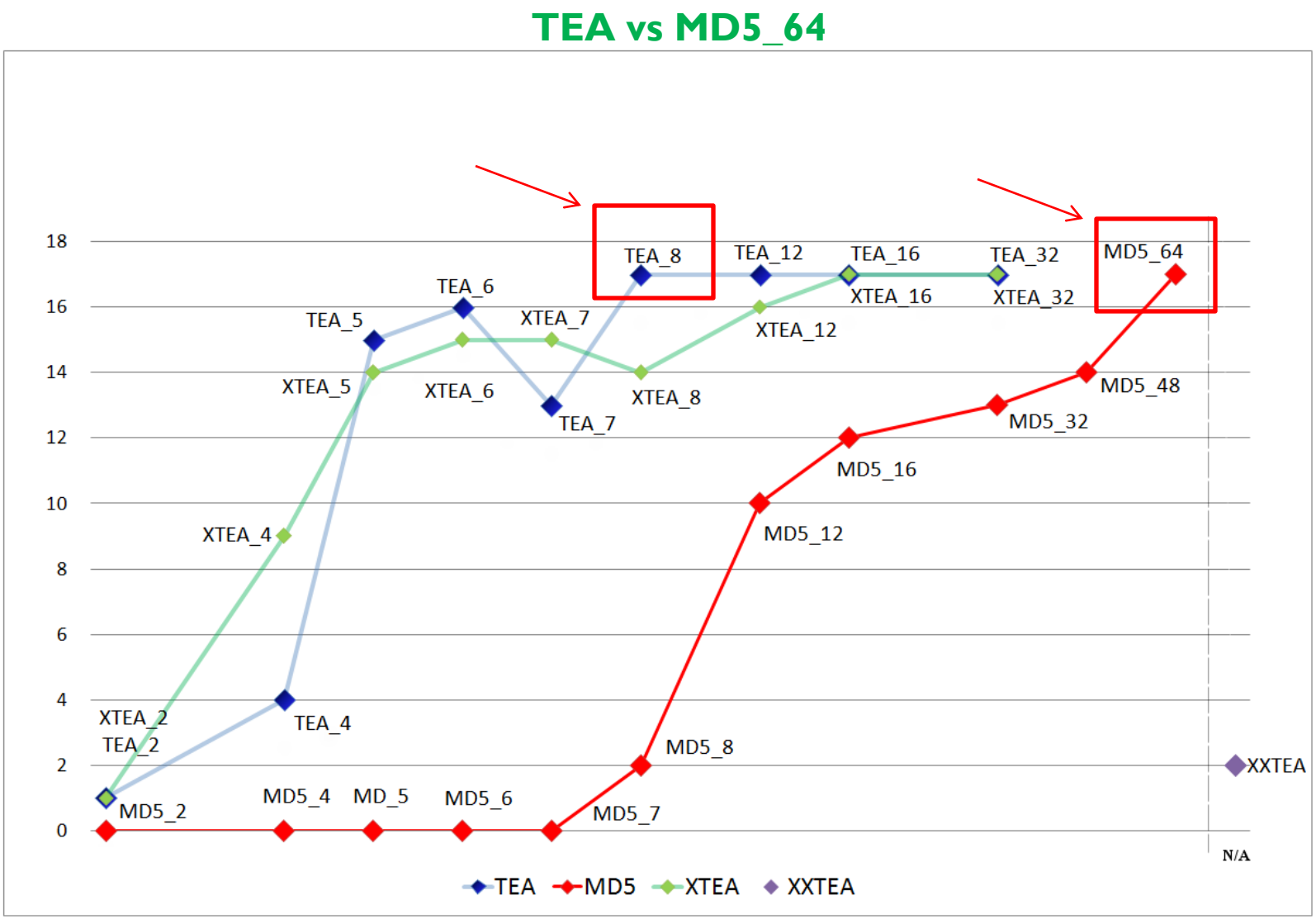
(d) $XTEA_1$ (e) $XTEA_2$ (f) $XTEA_3$

Per round time analysis for Encryption Algorithms as RNG

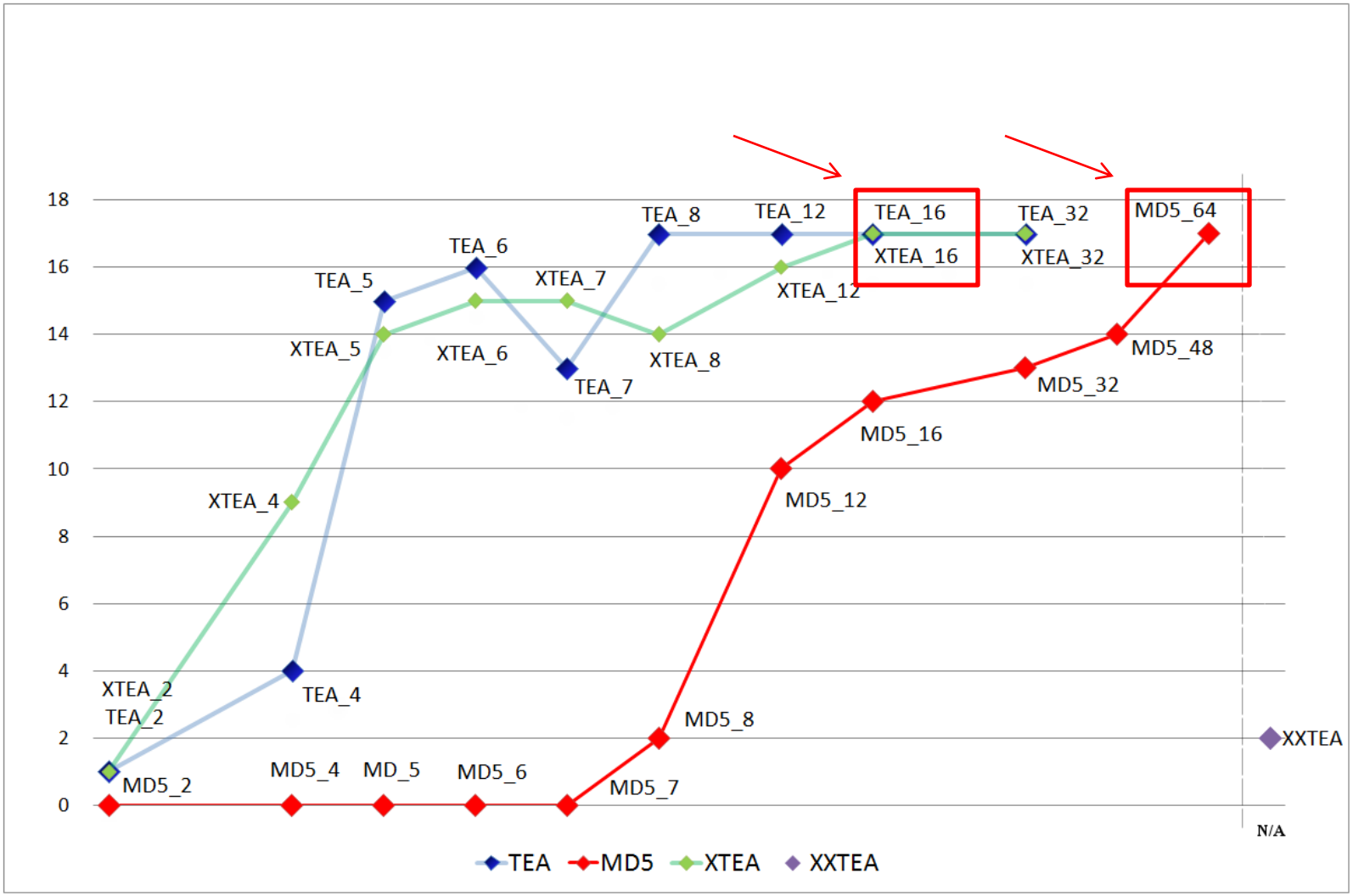


GPU: NVIDIA 9800 GT

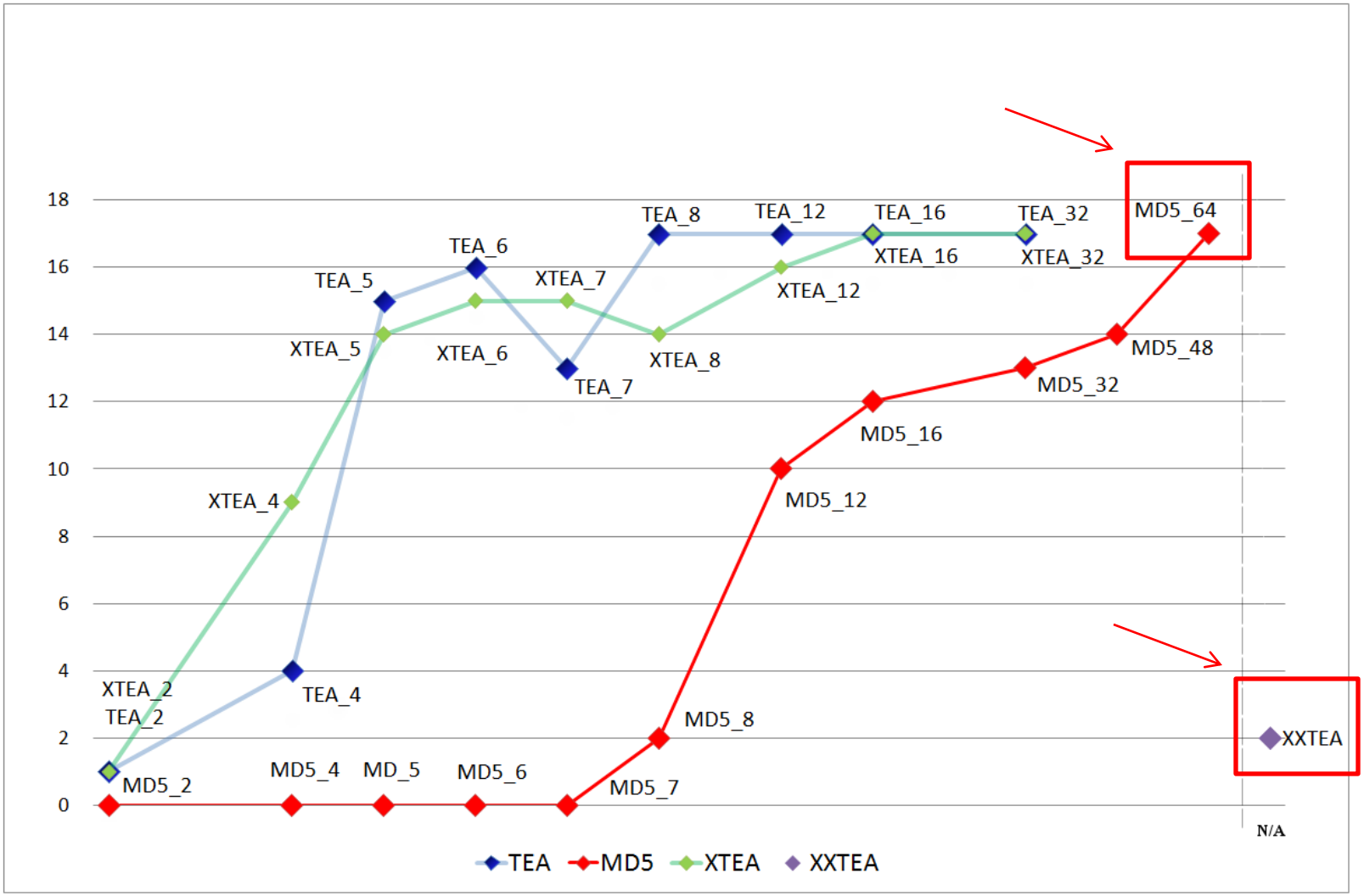
DIEHARD Tests passed



DIEHARD Tests passed

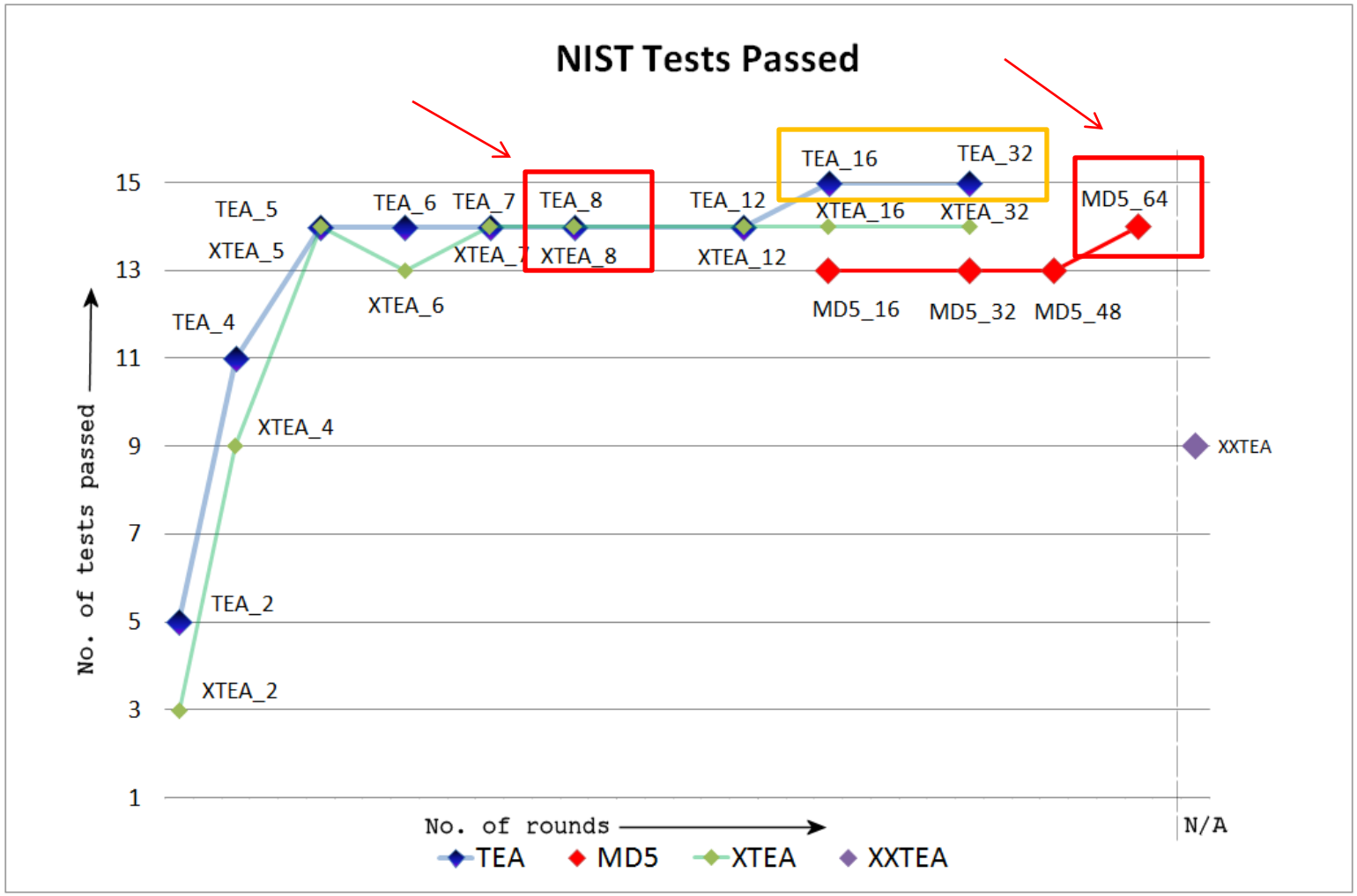


DIEHARD Tests passed

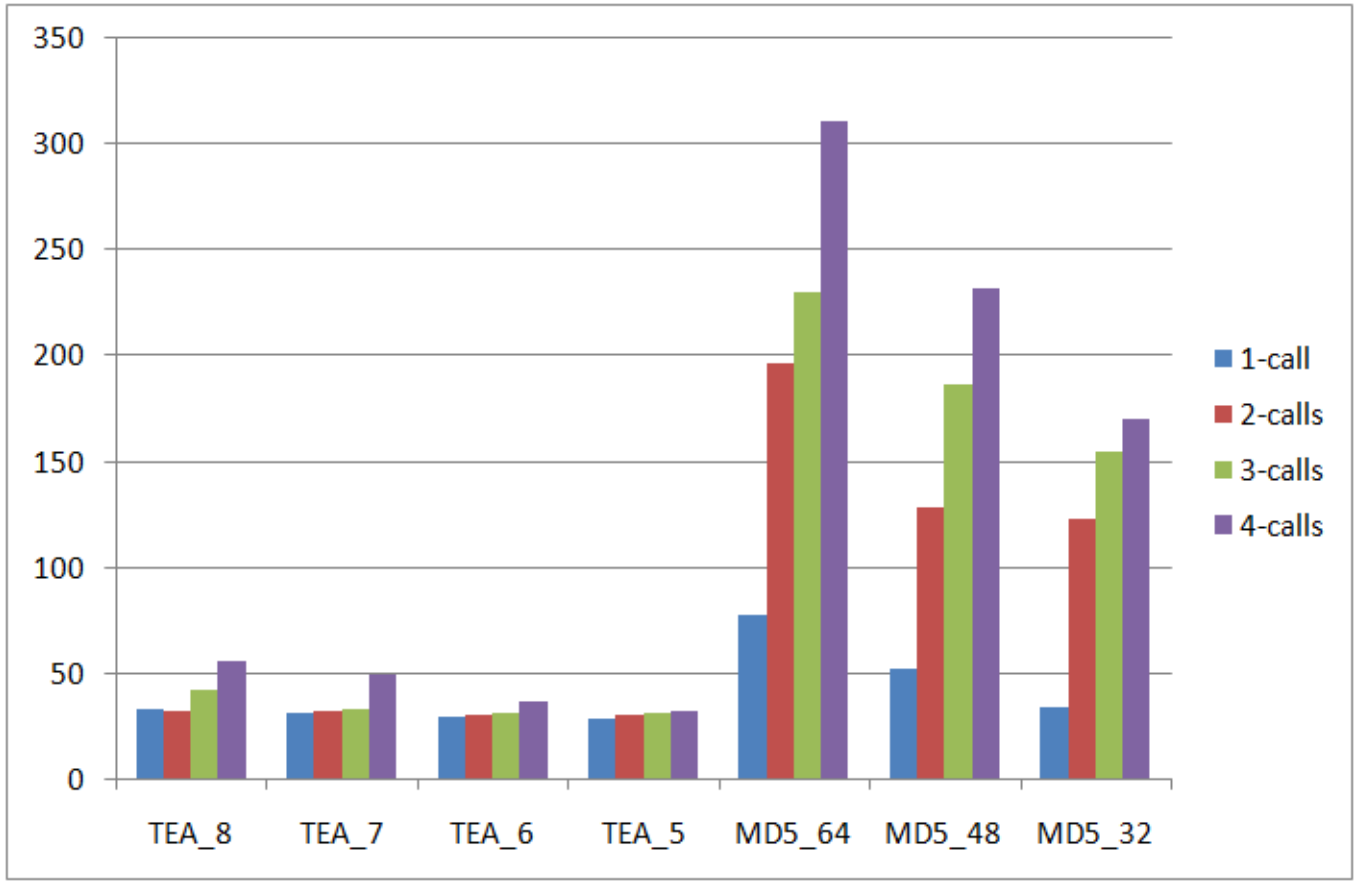


NIST Tests passed

TEA vs MD5_64

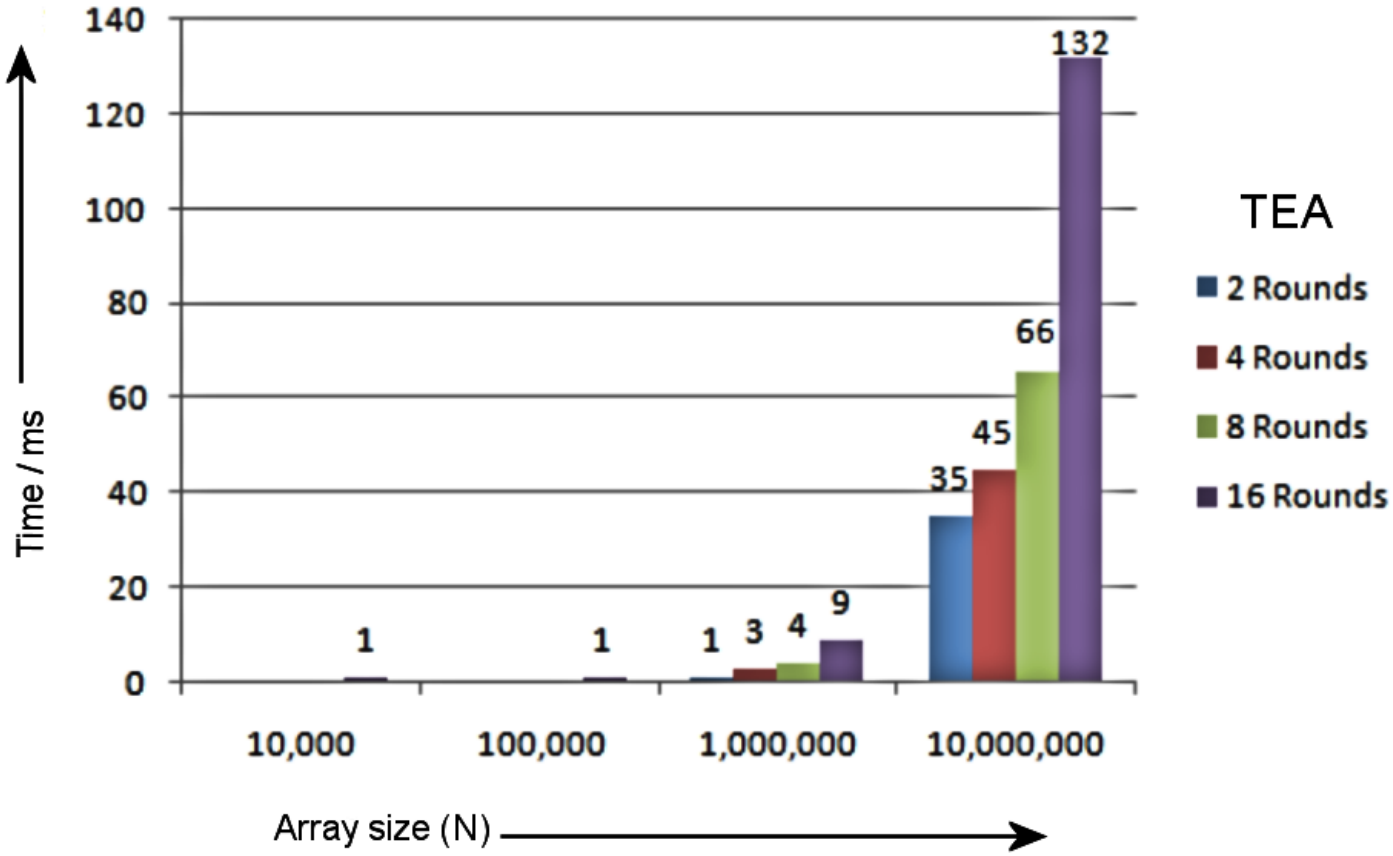


Usability scaling for PRNGs in shaders



GPU: NVIDIA 9800 GT

Cuda analysis results



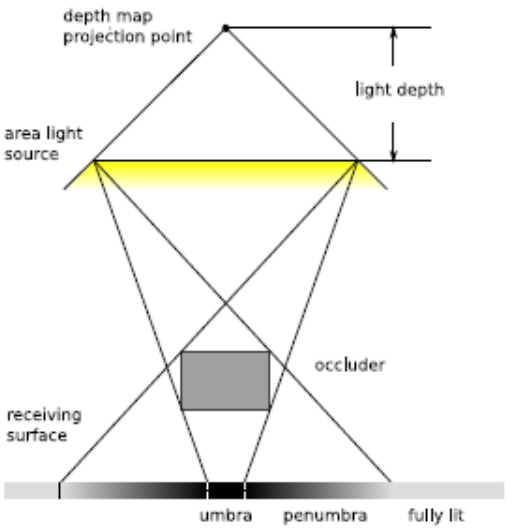
Time (in milliseconds) to complete all threads on NVIDIA 8600M. Some intervals were too short for the counter

Outline

- Introduction
- Previous Work
- Analysis and Results
- Monte Carlo Shadow Algorithm
- Conclusion

Real-time Soft Shadows via Monte Carlo Sampling

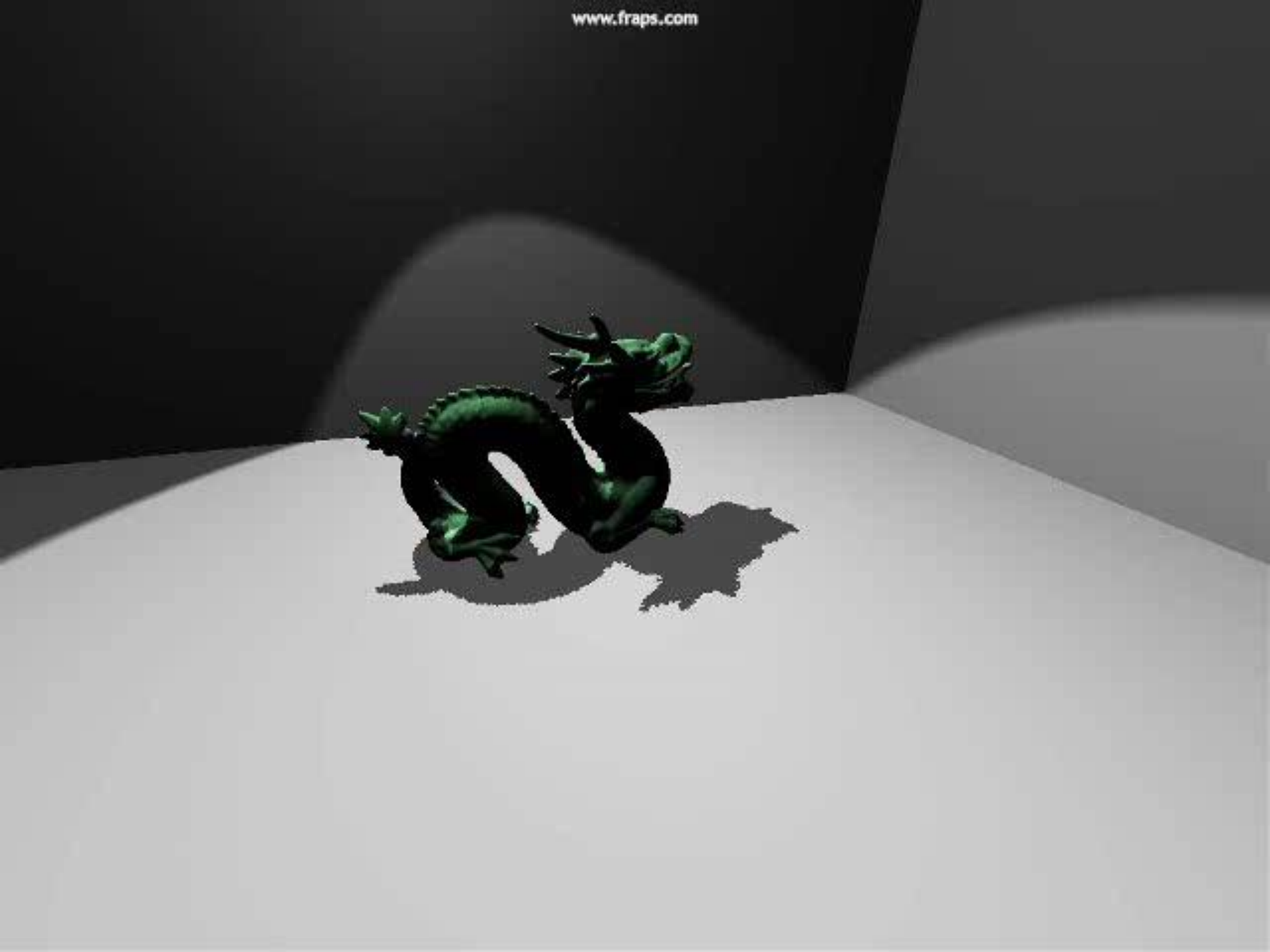
- Shadow maps are used to represent discretized occluding geometry
- More than one shadow maps.
- Ray tracing done through the depth maps.
- Depth sensitive Gaussian filter



An area light source and corresponding shadows. In our approach, the depth map is rendered from a projection point placed behind the center of the light.

Monte Carlo Shadow Algorithm demo

1. Algorithm Demonstration. Sampling at the edge of the shadow region
2. Increase Penumbra region
3. TEA-1
4. TEA-2 and TEA-4
5. Add filteration



Outline

- Introduction
- Previous Work
- Analysis and Results
- Monte Carlo Shadow Algorithm
- Conclusion

Conclusion

- We have provided an extensive per round analysis for TEA and MD5
- TEA is a better alternative to MD5 for GPU noise and as a PRNG

Conclusion

- Easily adjustable Speed/Quality tradeoff. Our approach is more tunable and comparatively better.

- TEA is easy to scale and implement
  - Adjusts to both compute intensive and compute extensive applications

Questions