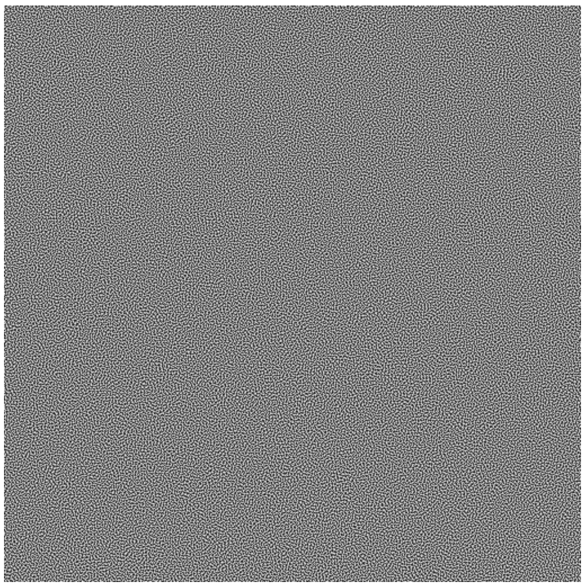
Preference of blue noise in ray tracing

Is blue noise visually preferable to white noise as ray tracing error distribution?



Thesis for the Master's degree of
Game and Media Technology

J. Vreugdenhil (6211046)
Supervised by P. Vangorp and A. Telea

Utrecht University
Netherlands
April 2024

Abstract

Blue noise is generally considered as preferable to other types of noise in graphics. However, there is a lack of empirical evidence for this claim. With this research, we aim to test this claim and provide additional understanding of the subject. We discuss literature on blue noise, sampling theory, and human vision to explain possible reasons for the preference of blue noise. We have performed an experiment to test whether rendered images with error distributed as blue noise were found to be visually preferable to images with error distributed as white noise. We found that there is a clear preference for blue noise over white noise when the amount of samples per pixel is equal. We also found that a preferable noise type does not outweigh the visual improvement of rendering an image with two samples instead of one.

Contents

1 Introduction

Blue noise has had a place in graphics for a long time, and its usage in ray tracing also dates back to 1987 (Mitchell, 1987). Due to stronger hardware and dedicated ray tracing cores (NVIDIA, 2023), ray tracing is becoming more common for consumer computers. Ray tracing being feasible in real-time nowadays makes it usable in video games. The real-time restriction does limit ray tracing to a few samples per frame, so it is important to make each frame count.

Blue noise is used in ray tracing to improve the perceptual quality of renders. It is generally viewed as more visually pleasant than other types of noise. However, there is a lack of empirical research for this claim. There also does not seem to be a conclusive explanation why blue noise is visually preferable to white noise. Blue noise is widely used in graphics and is the subject of much research. It is therefore important to test the validity of the claim that blue noise is visually preferable and to research possible reasons for this. Improving our understanding of the preference for blue noise may also help with further developments for visual media.

In this research we tested whether blue noise is perceived as visually preferable to white noise. We answered the research question: *"Is a rendered image with the error distributed as blue noise visually preferable over a rendered image with the same amount of error distributed as white noise?"* To answer this question, we conducted an experiment in which participants were asked to rank rendered image on visual pleasantness.

This thesis starts with a literature study, in which some concepts relevant to this research are explained and existing research on blue noise and perception based sampling is discussed. Then the research methods and experimental setup are explained. This is followed by and analysis of the results. Finally, the conclusion summarises the most important findings and discusses limitations of this research and opportunities for future research.

2 Literature study

2.1 Important concepts

In this section we introduce some concepts that are crucial to understanding the literature that is discussed later.

2.1.1 Ray Tracing

Ray tracing is a technique to draw, or 'render', virtual objects to a screen. Essentially it works by tracing light paths in reverse.
So, instead of light rays bouncing off an object and into a camera, the paths start from the camera and hit an object. Once it hits an object, the incoming light at that point has to be determined. This can be done by a very simple approximation if one assumes the object only receives direct light from a single light source. Then only the angle and distance to the light and whether there is an object casting a shadow in the way are taken into account. If the object is specular, like a mirror, one could bounce the ray again and take the shading information from the next hit. This is relatively quick but not fully realistic.
In the real world, any surface scatters light into multiple directions. To simulate this, one would send multiple rays from the object that was hit into other objects and let them bounce forever until they hit a light. By keeping track of all objects that were hit, one can realistically recreate how much light reaches the original object. Though this is practically unfeasible as the rays could have millions of bounces around the virtual scene before hitting a light source, and splitting into multiple rays at every bounce will make even the best computer run out of memory.
So, most ray tracing algorithms take a random selection of samples instead of simulating each light ray. If enough random rays are simulated, it will start to approximate the real world. This way of approximating is called Monte Carlo sampling. Monte Carlo sampling is a simulation technique

that is not unique to ray tracing. It is used in statistics when problems are impossible to solve analytically and taking random samples allows us to approximate a function.

2.1.2 Noise

When taking hundreds of samples, this approximation is fine and produces good looking results. In real-time applications, however, only 4-8 samples per pixel are typically taken. This can cause a large amount of visual noise, as pixels that are next to each other may have sampled the scene very differently.
Say, for a 1-sample-per-pixel example, there would be two pixels next to each other whose rays hit the same object. Both rays have a fifty percent chance to hit the light. One pixel by chance had its ray bounce directly into the light and thus is highly illuminated. The pixel next to it could have its ray bounced into a shadowy corner of the scene and thus be quite dark. Ideally you would have wanted both pixels to be half illuminated. Even though both pixels had the same chance of hitting the light, they received very different results. This is the noise that is introduced with Monte Carlo ray tracing when working with a low amount of samples.

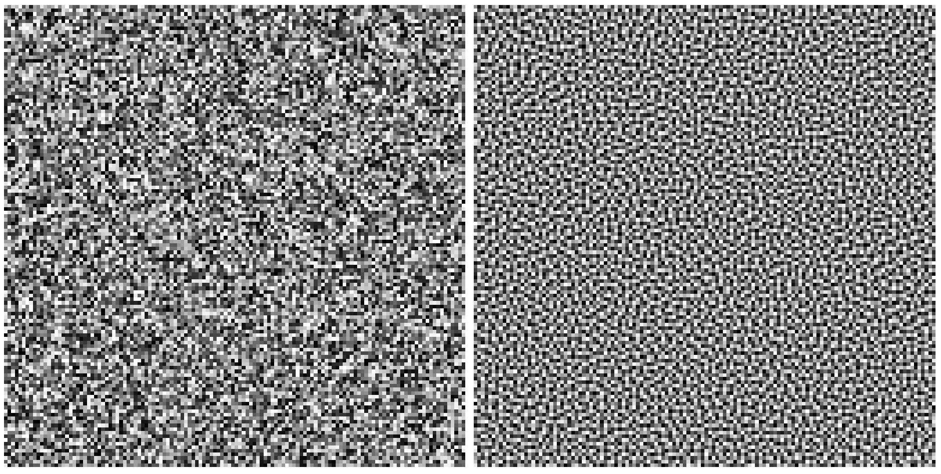

(a) white noise (b) blue noise

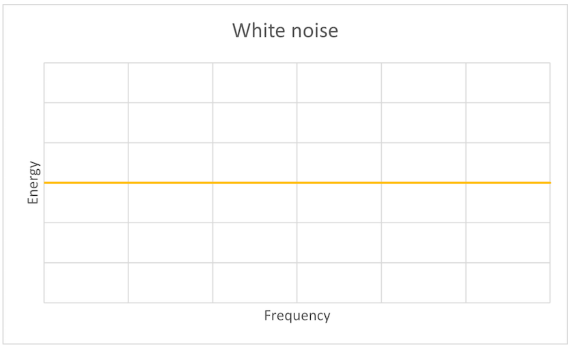
Figure 1: White noise and blue noise

2.1.3 White Noise

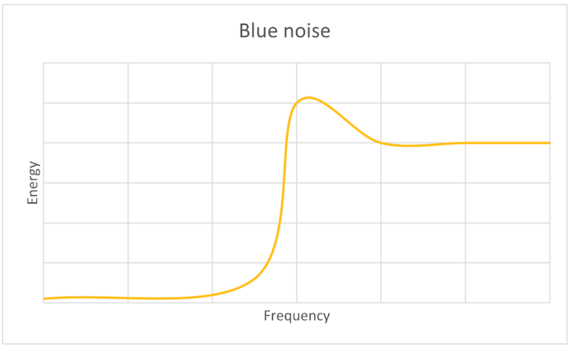
White noise is the name given to a signal where each value is random and fully independent of each other. Random Number Generators generate white noise. An image of spatial white noise can be seen in Figure 1a. The energy spectrum of white noise is shown in Figure 2a. You can see that there is an equal energy for all frequencies.

2.1.4 Blue Noise

The term 'blue noise' was coined by Ulichney in his book on digital halftoning (Ulichney, 1987). The term refers to an even, isotropic, yet unstructured distribution of points. This means the points are spread over the area in a way that does not form large clusters or empty areas, but also has no repeating structure. The isotropy requirement means the distribution has the same properties in all directions. According to Ulichney, the points are perceptually free of any structure. They call this a "grid defiance illusion". An image of spatial blue noise can be seen in Figure 1b. The power spectrum of blue noise (Figure 2b) looks like the power spectrum of white noise at high frequencies, but with minimal low-frequency energy. It also has a peak at the principal frequency of the pattern. This peak marks the transition from low energy frequencies to high energy frequencies (Ulichney, 1987) Blue noise became popular rapidly in computer graphics as its absence of structure prevents aliasing (De Goes, Breeden, Ostromoukhov, & Desbrun, 2012)



(a) White noise



(b) Blue noise

Figure 2: Energy spectra of white and blue noise

2.2 Benefits of blue noise

In graphics, incorporating blue noise is seen as a way to make images more pleasant to the eye. There have been several different claims as to why using blue noise reduces unpleasant artefacts. Broadly these are: Blue noise can reduce aliasing, the lack of structure is more pleasant to the eye, and high frequency details are more difficult to see. These three aspects are discussed in sections 2.3 to 2.5.

2.3 Aliasing

Transforming a continuous signal into a discrete signal is called sampling. Examples of this are:

- Digital photography, where light in the environment is turned into pixels.
- Digital Music recording, where a sound signal is sampled and stored in an audio file.
- Image rendering, where digital light in a digital environment is turned into pixels

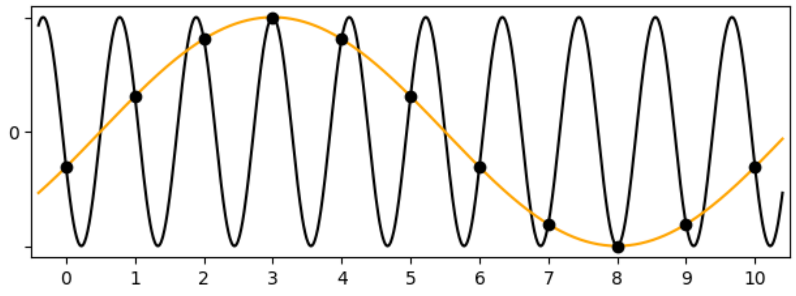


Figure 3: example of aliasing in sampling a sinewave

When the sampling rate (how many samples per second or samples per view angle) is too low, errors can occur.
As long as the sampling rate is equal to twice the maximum frequency of the signal that is being captured, no data will be lost. This sampling rate is called the Nyquist rate (Landau, 1967).
Conversely, the frequency that is half the sampling rate and thus will always be sampled correctly is called the Nyquist frequency. If the sampling rate is lower than the Nyquist rate, data can be lost and artefacts can be introduced. This is know as aliasing.

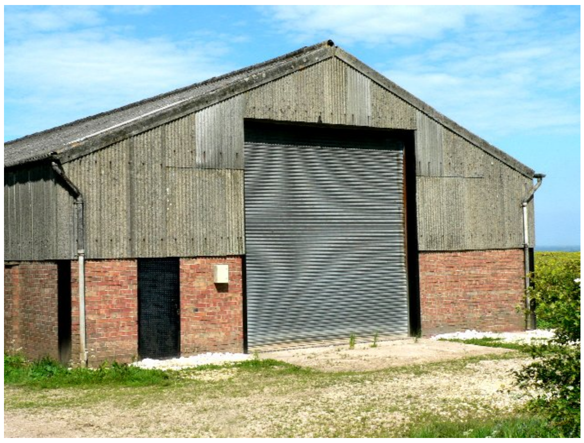
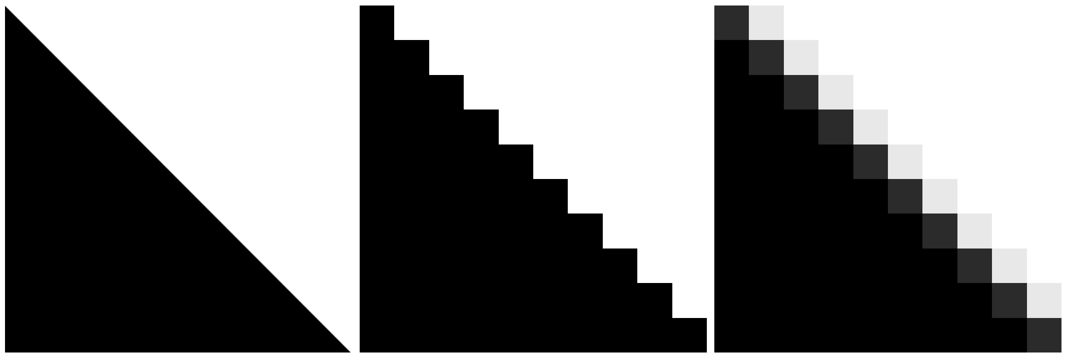


Figure 4: "Moiré Pattern at Gardham Gap", by Roger Gilbertson

A common example of aliasing can be seen in Figure 3. Here, the high frequency black sine wave is sampled at a too low frequency, displayed by the black dots. Reconstructing the signal based on these samples results in the orange function, which is very different from the original.
An example of this happening to a photograph can be seen in Figure 4 which displays an artefact known as a 'Moiré pattern' on the barn door. The pattern on the door has a high frequency which could not be accurately captured on a digital camera.
An other common occurence of aliasing are 'jaggies' at sharp edges. In Figure 5a you see a triangle with a sharp edge. In Figure 5b you see the same image but sampled at a very low rate. Each pixel (sample) is colored black if it is inside the black triangle and white if it is not. This causes a jagged edge to appear. In Figure 5c, an anti-aliased version is shown. Here, pixels are displayed in a grey values based on how much of the triangle is in the pixel, giving a slightly smoother edge. This can be done by taking many more samples inside a pixel and counting how many are inside and outside the triangle. This is called super-sampling and is a simple way to reduce aliasing effects.



(a) fullscale image (b) undersampled image (c) anti-aliased image

Figure 5: Anti-aliasing example

Blue noise in the retina

Yellott (1983) noticed that in human peripheral vision, no disturbing aliasing effects are present. Even when observing frequencies that are above human perception limits. They then analysed the distribution of photoreceptors in the retina of the eye of a rhesus monkey. These photoreceptors were found to be distributed in a semi-noisy pattern: it is neither perfectly regular nor perfectly random. A power spectrum of the noise showed that it contains almost no low frequencies, has a spike on a high frequency and then has equally distributed higher frequencies. This distribution of

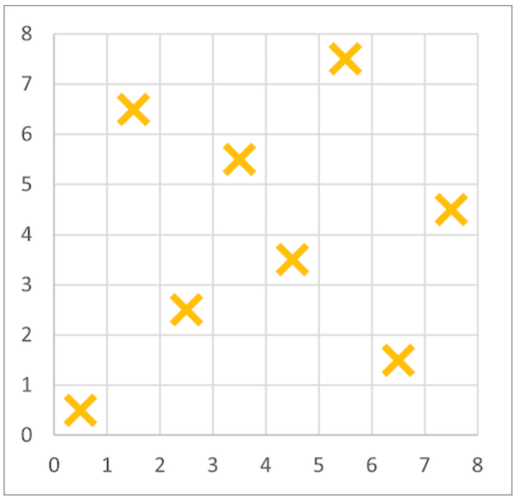
points would later be classified as blue-noise point distributions.
This distribution avoids introducing noise for low spatial frequencies (which would be under the Nyquist frequency, if the distribution were a regular grid) while higher spatial frequencies (which would normally cause artefacts patterns due to aliasing) are scattered into broadband noise, leaving no disruptive artefacts (Yellott, 1983).

Non-uniform sampling

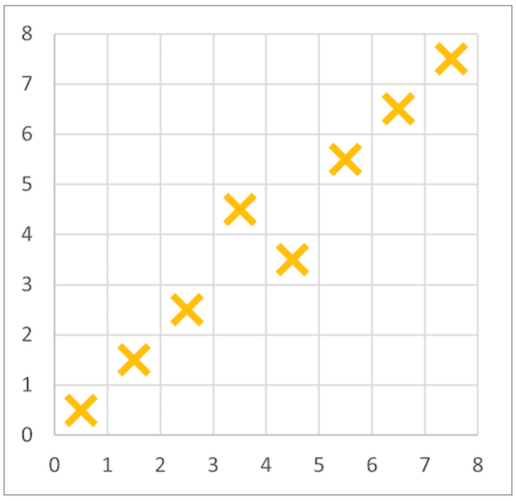
Mitchell (1987) implements the photoreceptor distribution in ray tracing. The photoreceptors follow a Poisson distribution (Ripley, 1977), which can be approximated by a *dart-throwing algorithm*. Mitchell proposes an algorithm that can generate these distributions on the fly, called the *point-diffusion algorithm*. Ray tracing samples are taken in a Poisson distribution over the screen, not accounting for pixel positions. Then these nonuniform samples are reconstructed into pixels using several filter passes. The research concludes that nonuniform sampling makes the aliasing noise less conspicuous to the viewer.

Raytracing without bouncing lights and with only a single light source is a 2-dimensional sampling problem. Distribution ray tracing adds motion blur, half-shadows from area-covering lights, glossy surfaces, and other lighting effects. This comes at the cost of sampling more dimensions. For example, motion blur can be simulated by sampling the scene at different times, which adds the time dimension. With these extra parameters, distribution ray tracing comes with a multidimensional sampling space.

Mitchell (1991) extends his previous work on nonuniform sampling to expand this to multidimensional sampling. A common way to do supersampling in ray tracing is stratified sampling. Here, each pixel is divided into subpixels. The two dimensions of the pixel, height and width, are each divided into N parts, resulting in $N * N$ subpixels. Within each subpixel, one random sample is taken and then all subpixels are averaged to construct the pixel value. For multidimensional sampling, each pixel is divided in more dimensions (D), for example time or depth can also be included. This means that in distribution sampling you would need to take N^D samples per pixel. Mitchell states that this could easily result in tens of thousands of samples per pixel. So for distribution sampling, only a selection of the N^D positions should be taken to keep computation times reasonable.



(a)



(b)

Figure 6: Two sampling patterns

Choosing which samples to take is important, as choosing correlated samples can introduce aliasing. In Figure 6, you can see two different two-dimensional sampling patterns. Both take an equal amount of samples and both cover the entirety of the two dimensions. However, in 6b you can clearly see a linear correlation between the sampling locations. This can cause aliasing, especially when dealing with diagonal features in an image. The uncorrelated sampling pattern 6a is much better at preventing aliasing.

This concept extends to sampling in multiple dimensions. Correlation between each dimension can happen. Even when only taking one sample per pixel, correlation between pixels should be prevented. Mitchell (1991) proposes an algorithm to generate good multidimensional sampling patterns that have a high spatial frequency. He sees that with this sampling pattern, the resulting noise in the rendered image is pushed into higher frequencies. Expert observers stated that an image with motion blur where the *time* value was distributed over the pixels by the new algorithm, looked better than the image where the *time* value was randomly distributed.

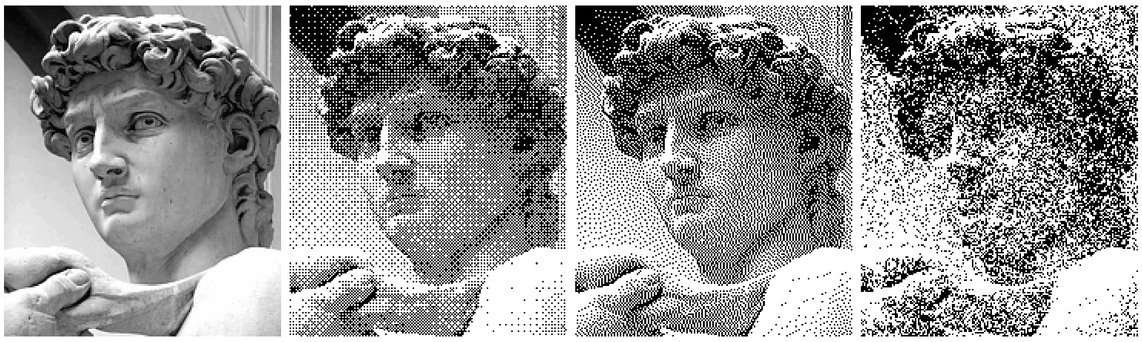
Georgiev and Fajardo (2016) improved upon this approach with their own algorithm. They generate multidimensional blue noise masks offline by starting off with white noise and swapping pixels to optimise for blue noise. Sample masks are 'images' that tell, for each pixel, where the other dimensions should be sampled. Because of these blue noise masks, neighbouring pixels evaluate very different locations in the sampling domain, yielding high-frequency noise in the rendered image (Georgiev & Fajardo, 2016), which was observed to be a desirable property (Mitchell, 1991).

Inspired by this work, Heitz and Belcour (2019) found a different way to distribute rendering error as blue noise. Starting off with a frame rendered as normal, the random seeds for the next frame are permuted locally to improve the error distribution. This is done by first sorting the pixels on luminance and then distributing them based on an example blue noise mask. This allows for a blue noise distribution of errors in consecutive frames with negligible overhead.

2.4 Lack of structure

Halftoning

Halftoning is the process of turning a greyscale image with many values into an image with either black or white points. This is used in black-and-white printers. With only black dots, it is possible to generate the illusion of grey tones by alternating black dots and white spaces.
For a grey tone that is exactly between black and white, a checkerboard pattern can be used. For many grey values between black and white, a constant pattern can be constructed. In this way, a greyscale image can be turned into a black-and white picture with different grey tones represented by different dot patterns. An example of this *ordered dither* can be seen in Figure 7b. A shortcoming of this ordered dither is that there is only a fixed number of grey levels for which a well-distributed repeatable pattern can be created. Note that, unless highly zoomed in on these images, they will most likely show up with aliasing artefacts due to the high frequency content.



(a) greyscale image (b) ordered dither (c) error diffusion (d) random dither

Figure 7: Examples of different dither algorithms, images dithered by Wikipedia user Gerbrant.

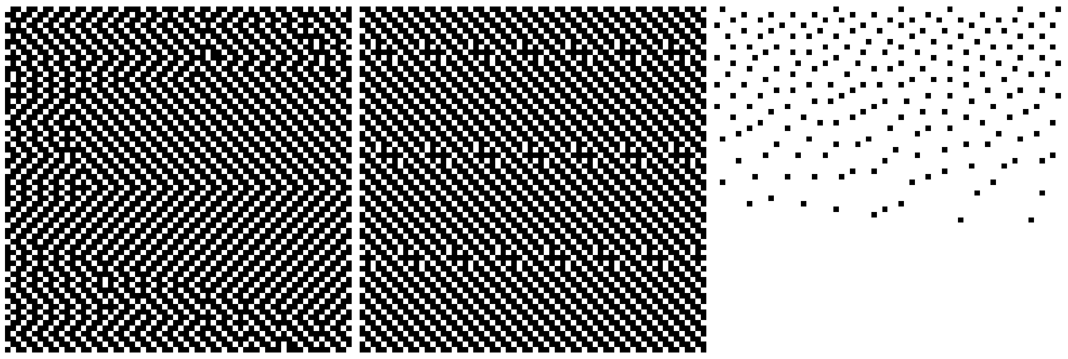
However, not all dot patterns are of equal quality. It is important that the pattern is directionless or 'isotropic'. If a pattern is anisotropic, it means that directional artefacts can be spotted, like dots placed in such a way that they start to look like lines. When the underlying pattern of the dots is too noticeable, it starts to clash with the pattern of the picture that it is meant to display. This is when it is useful to break up the pattern with random noise. Adding this noise is called dithering.

The Error Diffusion Algorithm

In Ulichney's book (1987) on digital halftoning, he addresses these problems in the chapter "Dithering with blue noise". The same information can also be found in his later released paper with the same title (Ulichney, 1988). In the book chapter, he experiments with adding noise to the halftoning process with the use of the error diffusion algorithm.
This algorithm, also known as the Floyd-Steinberg algorithm (Floyd, 1975) does not use predetermined patterns to perform the halftoning. Instead, it iterates over each greyscale pixel and assigns it as either black or white, depending on which it is closer to. The 'error' that is then generated by this is stored and distributed over pixels in its area, making the next pixels more biased towards the other colour value. This minimizes the total error of the image. An example can be seen in Figure 7c.

This algorithm is simple yet widely applicable, but some shortcomings are pointed out by Ulichney (1988).
The algorithm creates correlation artefacts for many grey tones. Correlation is when the pixels seem to follow a pattern, they have a certain 'correlation' with each other. This makes visible patterns emerge that are not in the original image. Examples can be seen in Figures 8a and 8b. Both of these images were solid grey images that were halftoned which produced diagonal line patterns.

It also creates artefacts due to the direction that the algorithm operates in, called directional hysteresis. This often introduces wiggly line-like artefacts in either very light or very dark areas, as can be seen in Figure 8c. Patterns introduced in an image that were not in the original greyscale image can be very distracting.



(a) correlated artefact (b) correlated artefact (c) hysteresis artefact

Adding noise

To break up any artefacts, Ulichney (1988) introduces noise to the halftoning process. He introduces the idea of white noise dithering. This is quite a simple algorithm. Each grey value is compared to a random threshold and is assigned either as a black or white pixel, depending on whether its value is above or below the threshold. The random values are supplied as a white noise map, meaning uniformly distributed and uncorrelated noise. An example can be seen in Figure 7d. This algorithm did not create any good images, but it did provide some valuable information: Ulichney states that while white noise patterns do not suffer from the correlated periodicity of ordered dither, it contains low-frequency patterns that give the result a very grainy appearance. This makes the resulting image less pleasant to look at. In contrast, the ordered dither patterns, while containing correlated artefacts, do have a power spectrum with mostly high frequency patterns.

Blue noise

Ulichney (1988) proposes a point distribution that incorporates this high frequency power spectrum but does not have the correlation artefacts of ordered dither. A pattern that is somewhere between ordered and random. This is called blue noise. The paper then shows various different ways of incorporating noise into the error diffusion algorithm, in order to obtain a blue noise distribution in the resulting image. Ulichney concludes that a well-formed dither pattern should not have its own structure but should be innocuous.

> "Blue noise is visually pleasant because it does not clash with the structure of an image by adding one of its own or degrade it by being too 'noisy' or uncorrelated (Ulichney, 1988, p.75)."

Pink noise

Elaborating on the idea of blue noise being 'innocuous', the opposite would be pink noise or 1/f noise. So called because the power of the noise follows the curve of 1 over the frequency. So while blue noise has higher power in higher frequencies, 1/f noise has higher power in lower frequencies. In natural images, low frequencies are more prevalent than high frequencies (Field, 1987). Studies also found that, in music, listeners found 1/f noise more 'interesting' than either $1/f^0$ or $1/f^2$ noise, also called white or brown noise, respectively (Voss & Clarke, 1978). So while 1/f noise is 'natural' and draws attention, blue noise is not interesting and, importantly, not disruptive. It has minimal structure and does not interfere with whatever it is added to.

2.5 Human Visual System

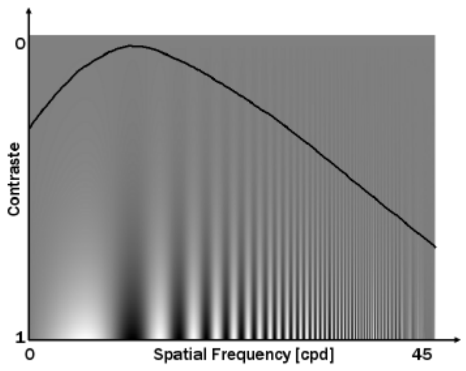


Figure 9: "Contrast Sensitivity Function (CSF) of the Human Visual System." by Jean-Philippe Tarel

Contrast sensitivity function

Next to being structureless and innocuous, blue noise is also likely to be less visible than other types of noise due to the way human vision works. Spatial frequency refers to the amount of brightness changes in a part of the field of vision. The eye is more sensitive to certain spatial frequencies than others. Very low or very high frequencies are perceived to have a lower contrast while the middle range is perceived to have a higher contrast. This is called the contrast sensitivity function (CSF) (Barten, 1989; Kelly, 1977; Lubin, 1995). A visual representation is shown in Figure 21.
As previously discussed, blue noise consists of mostly high-frequency noise. This fact combined with the knowledge that the human eye is less sensitive to high frequencies, could explain why blue noise draws less attention than other types of noise and is therefore pleasant to use in computer graphics.

Modeling the Human Visual System

The CSF is a core component of our understanding of the human visual system (HVS). Much research has been done into how we form a neural image of the outside world. The mechanics are simulated in HVS computational models. These models are often used in computer graphics for image compression, quality assessment and image enhancement (Granrath, 1981).
Simulating the HVS is very useful for graphics, as almost any graphical application is meant for human eyes. It allows for removing data that would not be perceived naturally, and focusing on what is visually important.
Sullivan, Ray, and Miller (1991) introduced a human vision model to generate optimal halftoning patterns which are minimally visible. This is mostly done by optimising for having high spatial frequencies in the patterns.

Visual Difference Predictors

Human vision models are often used in Visual Difference Predictors (or Visual Discrimination Models). These use the model of the eye to calculate how much difference a human would actually perceive between two pictures. This can be used in, for example, image compression. The compression method that causes minimal visual difference between the original picture and the compressed picture is often the best.
An example of finding the difference between images in a way that ignores the human visual system is the Mean Square Error (MSE) metric. In this calculation, each pixel is compared between the two images and the difference between the pixels is squared and summed. This is an

easy and fast way to calculate differences, but in terms of finding perceptual differences it is not very useful. This disconnect between absolute pixel difference and perceptual difference in images is important to the research of why blue noise is preferable, as incorporating blue noise in path tracing does not often improve the MSE value of an image. It merely improves the subjective quality (Georgiev & Fajardo, 2016).
Some early and influential visual difference predictors are the Daly VDP (Daly, 1992) and the Sarnoff VDM (Lubin, 1995). These can express the difference between two images as a single value that represents perceptual error.

Perception based adaptive sampling

Bolin and Meyer (1998) created an adaptive sampler that uses an image quality model that emulates the human vision system. Based on where there are visible artefacts, additional samples are assigned. They use a vision model that is heavily inspired by the Sarnoff model, but focuses on speed and efficiency as speed is quite important in path tracing.
The paper also describes noise masking. Noise masking comes from the heightened sensitivity to certain spatial frequencies. In a noisy image, in areas where there is a frequency present that the eye has a strong sensitivity to, it masks the noise, making it less noticeable. As an example, in a natural image, noise would be more visible on a clear sky than on a highly textured rock. This can be used to concentrate more samples on areas where noise would be more visible.

Chizhov, Georgiev, Myszkowski, and Singh (2022) also use a perceptual error model to distribute samples and spread out pixel error in a way that minimizes perceptual error. They bridge the gap between halftoning literature and error reduction in path tracing. Using halftoning techniques such as error diffusion and ordered dither, they aim to reduce perceptual error in path tracing. In contrast to many perceptual error reduction techniques, this algorithm can also decrease the total image error. Meaning it can improve the renderings not only subjectively, but also numerically. They note that when optimizing for the human visual system, the algorithm naturally produces a blue noise distribution of the error as a by-product. With a visual experiment they demonstrate how a blue noise distribution of rendering error reduces the visibility of that error. By emulating an increased viewing distance with a gaussian blur kernel, they demonstrate that blue noise converges to a constant image much faster than white noise.

Pappas and Neuhoff (1999) found that the gaussian is a good approximation of the Point Spread Function (PSF), which models the light scattering in the human eye (Mantiuk, Daly, Myszkowski, & Seidel, 2005).
Chizhov et al. (2022) states that high-frequency noise becomes indiscernible at viewing distances where actual image detail, which is mostly low or medium frequency is still visible to the human eye. So at the viewing distance where it becomes impossible to notice the blue noise, the rest of the content should still be visible.
For white noise, you would have to look from a much further distance in order to not see the noise anymore. At that point, any high or medium frequency content from the image is also invisible.

Chromatic noise

While contrast in graphics often references to just the brightness value of colours, chromatic contrast describes the contrast between different hues. This is processed by different mechanisms in the human visual system than achromatic contrast and thus behaves differently.
In general, the eye is less sensitive to colour differences than brightness differences, requiring higher chromatic contrast for the same amount of visibility as achromatic contrast. For low frequencies though, there is no drop in sensitivity compared to medium frequencies. This is very different from the achromatic CSF in which the sensitivity for low frequencies is less than for medium frequencies (Bolin & Meyer, 1998).

Johnson and Fairchild (2005) performed an experiment on the visibility of different frequencies of chromatic noise and luminance noise. Participants were asked to identify which images with

overlaid noise they deemed to have the highest image quality. The least favourite images would be the ones where the noise is most visible.
Five levels of spatial frequency were tested: 2, 4, 8, 16 and 32 cycles-per-degree. They concluded that achromatic noise is more visible than chromatic noise overall. For chromatic noise, the lowest spatial frequency (2 cycles-per-degree) was most visible. For achromatic noise, the most visible noise was the second-to-lowest frequency: 4 cycles-per-degree.
To illustrate, if you were looking at a screen from 50 centimeters away, 4 cycles-per-degree would be about 2.18 millimeters while 2 cycles-per-degree would be about 4.36 millimeters. The highest frequency tested was 32 cycles-per-degree which would be equal to about 0.27 millimeters. The theoretical maximum frequency humans can see is yet unclear, with claims ranging from 40 to 60 cycles-per-degree. A visual acuity of at least 30 cycles-per-degree is considered standard. This corresponds to having 20/20 vision, or 6/6 vision in Europe (Holladay, 1997).

Temporal frequency

Much like the spatial contrast sensitivity function, there is also a temporal contrast sensitivity function. It takes the form of a bandpass filter, peaking at 5hz and with a high-frequency cutoff at 30hz (Granrath, 1981). Interestingly, chromatic contrast in the temporal CSF differs in the same way as in the spatial CSF. For chromatic contrast, the temporal function is a low-pass filter just like for the spatial function.

There is a rendering implementation that uses spatiotemporal blue noise as a rendering mask (Wolfe, Morrical, Akenine-Möller, & Ramamoorthi, 2022). This is a three-dimensional distribution that follows the blue noise principles through all dimensions. This means that the samples at different times for the same pixel are also decorrelated. This decreases the amount of necessary samples because the samples are more evenly distributed over the sampling space. These masks are blue in the temporal domain, meaning they should also have a higher temporal frequency than temporal white noise. Combined higher spatial and temporal frequency might push this noise even further out of the visible range when used in a 1-sample per frame setting.

3 Methods

3.1 Introduction

According to an influential early paper on optimal sampling patterns for distribution ray tracing, an image rendered with one sample per pixel using a blue noise distribution is visually preferable to an image with three samples per pixel with white noise (Mitchell, 1991). When studying the effects of non-uniform sampling of the time dimension for an image with motion blur, the researchers state that "The consensus was that three or four random time samples per pixel were required to match the subjective quality of (the one sample-per-pixel non-uniformly sampled image)"(Mitchell, 1991, p.161).
This is an observation from a small group of people in a specific experiment: Several expert observers comparing one image against a set of images with different amount of samples per pixel. Apart from this, there is a lack of information on the subjective preference of blue noise. There is much research towards optimally generating blue noise and applying it in ray tracing samplers. It has been proven that blue noise improves rendering convergence rates (Wolfe et al., 2022) and that it helps reduce aliasing (Ulichney, 1988). However, blue noise itself is also often stated to be visually pleasant, while there is no empirical research done toward that claim.

That is why we research whether blue noise is seen as visually preferable to white noise, when not considering the other benefits of blue noise. We also test the previously mentioned claim that one-sample blue noise is comparable to white noise with three or four times the amount of samples taken.

3.2 Research question(s)

The research question we want to answer is the following:

Is a rendered image with the error distributed as blue noise visually preferable over a rendered image with the same amount of error distributed as white noise?

3.3 Variables

To answer the research question, we set up an experiment in which participants were asked to rank different images on visual preference. The images to be ranked are computer renders of the same scene but with different noise parameters.

Several variables are identified as relevant in this experiment. The independent variables are:

- Noise type: This is categorical, either blue noise or white noise.
- Sample amount: A numerical variable, either one or two samples per pixel.
- Scene: A categorical variable that determines which scene the image depicts.

The dependent variables are:

- Total Error: The total error of any stimulus image.
- Perceptual image quality: The ranking that participants give a stimulus image. Ranges from 1-4, with 1 being the highest quality and 4 being the lowest quality.

In the analysis of the data, the noise type and sample amount are grouped together to form the four 'noise types': *1 spp white*, *1 spp blue*, *2 spp white*, and *2 spp blue* noise. All variables will be explained further in this section.

3.4 Experiment

For the experiment we generated ten sets of stimulus images. Each of these sets depict a different scene. The sets consist of four images each: two images with a white noise error distribution and two images with a blue noise error distribution. The images with the same error distribution are different in the amount of samples that were took per pixel to generate the images. We ended up with these four image types: 1 spp (sample-per-pixel) white noise, 2 spp white noise, 1 spp blue noise, and 2 spp blue noise.
These stimulus images are described in more detail in section 3.4.1

The images were included in an online survey which participants could complete independently. They were first explained the experiment and which data would be collected. Then they were asked for consent in participating in the research. If consent was given, they were presented with a short questionnaire. The participants were asked about:

- Their age
- Whether they have normal or corrected to normal vision
- How much time they spend looking at screens.
- On what kind of screen the participant is completing the survey

The reasoning behind these questions is discussed in section 3.4.5.

After completing the questionnaire, participants were taken to the actual experiment. For each image set, the four images were displayed side-by-side. The participant was asked to rank these images on visual pleasantness by dragging them to the place on the ranking they think it belongs. The ranking went from 'most pleasant' on the left side to 'least pleasant' on the right side. An example of this can be seen in Figure 10.

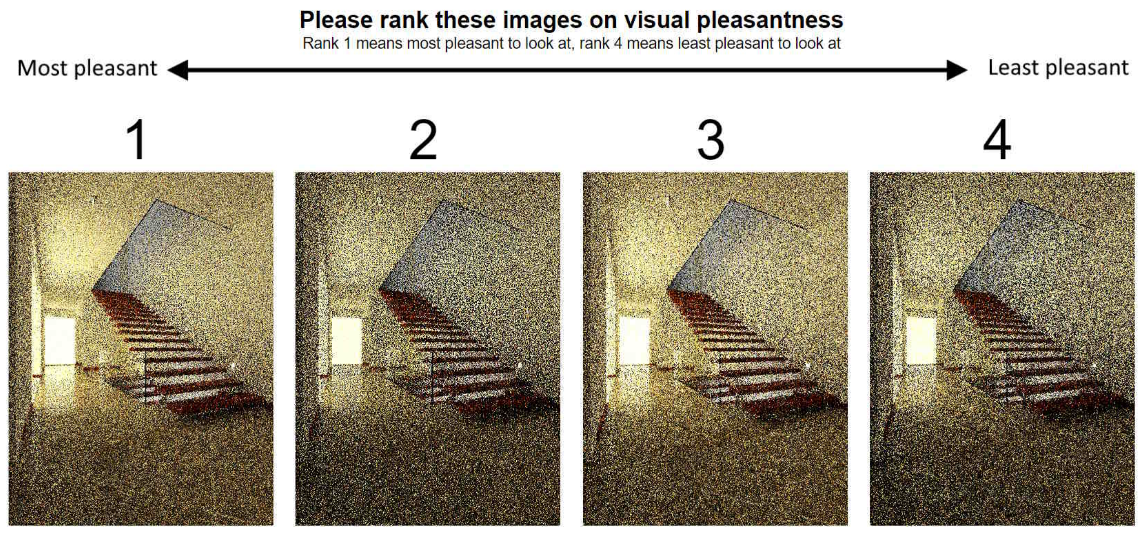

Please rank these images on visual pleasantness
Rank 1 means most pleasant to look at, rank 4 means least pleasant to look at
Most pleasant
Least pleasant
1
2
3
4

Figure 10: Example of a ranking task

The total experiment consisted of ten of these image sets. When completed, the participants were taken to a final page confirming that the experiment was finished and thanking them for their time.

3.4.1 Stimulus images

Creating the stimulus images was a crucial part of the experiment. This required great attention to detail because the small differences in the images were the subject of the research. Any noise or artefact that was not intentionally created, could cause problems in the experiment.
We used images that were identical to each other except for noise type, to ensure a fair comparison. The images also had to have clearly visible noise so that it would be easier for participants to notice the differences. As there was no dataset of images that fulfils these specific requirements, we had to generate the images ourselves.

The images were generated using a physically based rendering engine. This engine had to be able to use different sampling types for the light calculations so that we could create both blue noise and white noise. The engine also had to have the option to turn off any denoising or post processing so that the noise was not reduced or changed. The last requirement for the engine was that it should be able to run on the available hardware, which meant that using RTX cores was not possible. With all these requirements, the best engine to use was PBRT (Physically Based Rendering Toolkit).

PBRT, version 4, is an open-source rendering toolkit that is described in the book *Physically Based Rendering: From Theory to Implementation* (Pharr, Jakob, & Humphreys, 2023). This book explains the theory behind physical rendering and also provides all practical information needed to design a rendering system. The source code for the rendering toolkit is available on github (Pharr, 2021). PBRT fulfils all requirements we had for choosing a renderer. The fact that it is open source and that there is an entire book explaining how it works, also provided more certainty that unforeseen problems would be fixable.

PBRT works with scene files that specify both the scene and the rendering parameters. One of the parameters that can be specified in these files is which ‘sampler’ is used. The sampler determines where light samples are taken, which determines the type of noise in the image. There are many samplers to choose from but we are interested in the ‘Independent’ and the ‘Zsobol’ samplers.

The independent sampler is a simple, random sampler that does not target a specific sample distribution. The authors state that it is the least effective and mostly only included for comparison purposes, which is what we used it for. The independent sampler does not distribute noise over pixels, because it samples independent of other pixels. This uncorrelated sampling where each pixel picks identically distributed samples, gives us images with white noise.

The zsobol sampler is more complicated. It is an implementation of a sampler that diffuses sampling error by a hierarchical ordering of pixels, introduced in a paper by Ahmed and Wonka (2020). This sampler is able to produce decorrelated samples without structured aliasing artefacts and it is able to do this quick enough to generate new sampling patterns for each frame. By using a Z-order curve, also known as Morton ordering (Morton, 1966), each pixel is assigned a unique number. The screen space is hierarchically subdivided and pixel indices are scrambled to break up the recurrent structure. These pixel indices are then used as indices for a low-discrepancy sequence, namely a sobol sequence. A sobol sequence is a predefined mathematical sequence with the property that for all values of N, its subsequence $x_1, ..., x_N$ has a low discrepancy. By combining a Z-order curve, scrambling, and a low-discrepancy sequence, this method provides a good diffusion of sampling error over the image without structured aliasing artefacts. This gives us images with a blue noise error distribution.

The pixel filter of the renderer determines how samples inside a pixel are filtered. The 'ray' that is sent through each pixel, is not always sent through the middle of the pixel. Each ray is sent through a different point in the pixel, determined by the sampler. The pixel filter determines how these samples are combined together. A 'box' pixel filter is equivalent to having no filter, each sample is weighed the same. A triangle filter makes samples in the middle of each pixel weigh more towards the resulting image than samples towards the edges of pixels. This is a way to smooth out the image and reduce aliasing.
For the image rendering, the pixel filter was already set to a triangle filter. Since this was set the same in each scene and it has little impact on the resulting image at these low sample amounts, we decided to use the triangle pixel filter. One of the ten scenes ('Sportscar') was taken from a different source and only after the experiment had been conducted, we found that this scene did not specify the pixel filter. This meant that the renderer would have chosen the default filter which is the 'gaussian filter'. This filter uses a gaussian bump to weigh the samples more smoothly than the triangle filter. After noticing this inconsistency, we compared images of the sportscar render with both triangle filter and the gaussian filter. We found that that the images had small differences in highlights and edges between the two filters, but no difference at all in the noise structure. This leads us to believe that the pixel filter has had little to no effect on our research.

Other settings were set in the scene files so that all images had the same parameters:

- The integrator implements the light transport algorithm that computes the radiance arriving from objects in the scene. This can range from a very basic random walk algorithm, without any explicit light sampling, to a volumetric path tracing algorithm. For this experiment, the normal 'path' integrator would work well. We are not using volumetrics in the scene, and bidirectional path tracing or photon mapping would only complicate the process without adding anything to the research. The path integrator does implement NEE (Next Event Estimation), which estimates direct light on surfaces to reduce variance. Without this, it would be very hard to see anything on images with one sample per pixel. It also implements MIS (Multiple Importance Sampling) which makes points on a surface more likely to sample light sources that would contribute more light to that surface. This decreases chances of 'wasted' rays that contribute little light information. Both NEE and MIS are features you find in any professional path tracer implementation.
- The maximum number of light bounces was set to ten. The number was chosen to be high enough that increasing it makes a negligible difference to the output and to be low enough to save computation time.
- The image size is set to 384x512 pixels. This size was chosen to make it possible to display four images next to each other in full resolution on most screens.

- All other settings were left to default. Camera settings like field of view and position were left as they were in the original scene files.

A drawback of using PBRT is that it it can only render from pbrt files. There is no 3D editor to view and change the scenes. There is also not a simple way to export scenes from 3D modeling programmes like Blender to PBRT. There is a tool that does this for the third version of PBRT, but we were using the fourth version for which this tool was still in development. This meant that it would have been difficult to create our own scenes or copy existing test scenes for this experiment. Luckily, there were some scenes that were made for PBRT available for download.

Of the ten scenes that were used, nine came from Benedikt Bitterli's (2016) collection of resources for rendering research. These scenes range from very small setups with one or two lights to complicated scenes with a lot of indirect lighting. We picked the nine scenes so that a range of scene complexities was displayed. We avoided picking scenes where advanced lighting effects like water caustics, glass refraction, or hair rendering were the point of interest as those would likely generate images with such high amounts of noise that participants could not recognize the intended scene with the low amount of light samples we were aiming for. The tenth scene came from the example scenes provided by the developers of PBRT in a seperate github repository (Pharr, 2020). This is the 'sportscar' scene. We have edited this scene slightly. The camera position and field of view were changed so that the car could be seen from the front to better fit the image dimensions. We also removed the background because it was distracting in this new viewpoint.

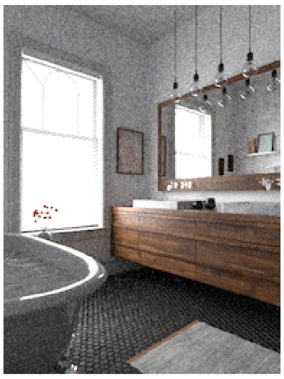
All scenes used in the experiment are shown in Figure 11. These reference images are rendered at 512 samples per pixel.

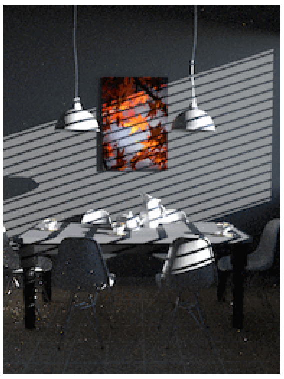
Using these scenes, we generated four versions of each image: 1 and 2 samples per pixel with both the independent sampler and the zsobol sampler. For brevity, these image types will be referred to as:

- **1 spp white**: 1 sample per pixel with the independent sampler.
- **2 spp white**: 2 samples per pixel with the independent sampler.
- **1 spp blue**: 1 sample per pixel with the zsobol sampler.
- **2 spp blue**: 2 samples per pixel with the zsobol sampler.

(a) 'Little Lamp'

(b) 'Victorian Style House'


(c) 'Contemporary Bathroom'

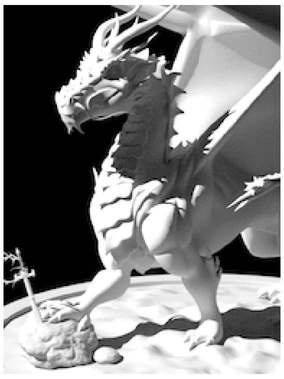

(d) 'The Grey and White Room'

(e) 'The Breakfast Room'

(f) 'Sportscar'

(g) 'The Wooden Staircase'

(h) 'Utah Teapot'


(i) 'Dragon'

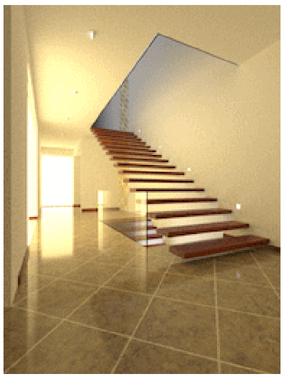

(j) 'Modern Hall'

Figure 11: Used scenes

We analysed the images on the amount of rendering error each one has. It is important that the total error of rendered images is not too different between the blue and white noise samplers. We want to test whether the distribution of the error makes an image more visually pleasant and for that we need the absolute error to be about equal to ensure a fair comparison. Due to the random nature of rendering, the absolute error will always have slight deviations and therefore will never be exactly equal. To test the absolute error we will use a simple calculation:
We take a high-sample reference image which will be seen as the ‘correct’ render. In our case, that is a 1024-sample rendering using the ‘zsobol’ sampler. Even at this high level of samples there is some noise left which could be fixed by rendering at higher sample counts. Due to hardware limitations and time restrictions we did not do that. For the purposes of this test, the reference images are good enough as we just want to compare the approximate rendering error of each image. With the reference images in place, we can find the error for each image by calculating the sum of the pixel-by-pixel difference between render and reference.

These error statistics are plotted in Figure 12. Here we see that for each scene the *1 spp blue* and *1 spp white* images are very close in total error amount. That means that it will be a fair comparison. The two-sample images naturally have a lower error amount, as more samples make the image closer to the correct image. It is worth noting that the two-sample zsobol images have a smaller total error than the two-sample independent images. That is because the zsobol sampler also decorrelates successive samples within the same pixel. Because the two samples are better distributed over the pixel, there is overall less error. This is a separate benefit of blue noise but it is not the subject of this research. Because of the error difference between the 2-sample noise types, it is more fair to use the 1-sample images for conclusive comparison. Further analysis of the images is discussed in the results section.

Examples of the different image versions are shown in Figure 13. Once we had these images, we could start ranking them.

3.4.2 Task design

In the online survey, the participants were asked to rank the four different images on visual preference. They could do this by dragging the images towards the place on the ranking they think the image belonged. We chose for this drag and drop method because it makes tasks more fun and improves the participants’ engagement. This makes for more accurate results and less chance that participants get bored and leave the experiment (Cunningham & Wallraven, 2011). For this task, it also seemed the most intuitive way for participants to do ranking.

We decided to make the task a ranking task between four options for a number of reasons. First, ranking allows for many implicit comparisons to be made between all images. For this research question we want to compare two different types of noise and two different types of

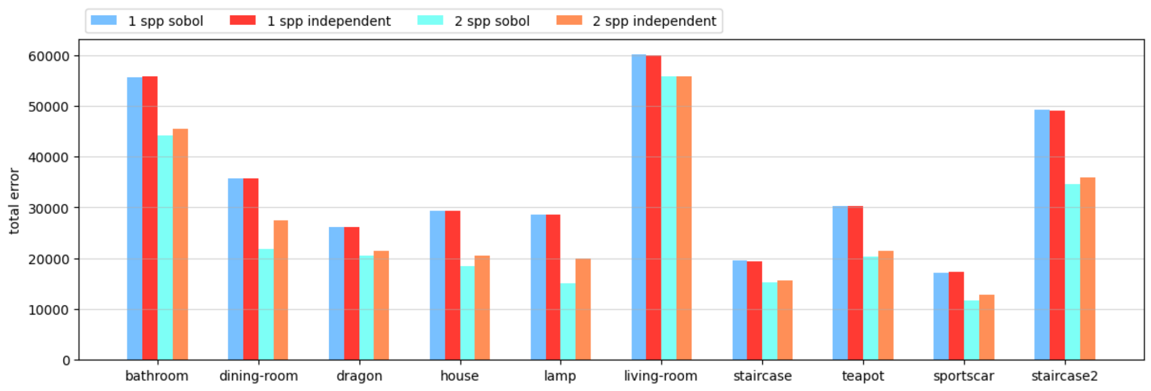
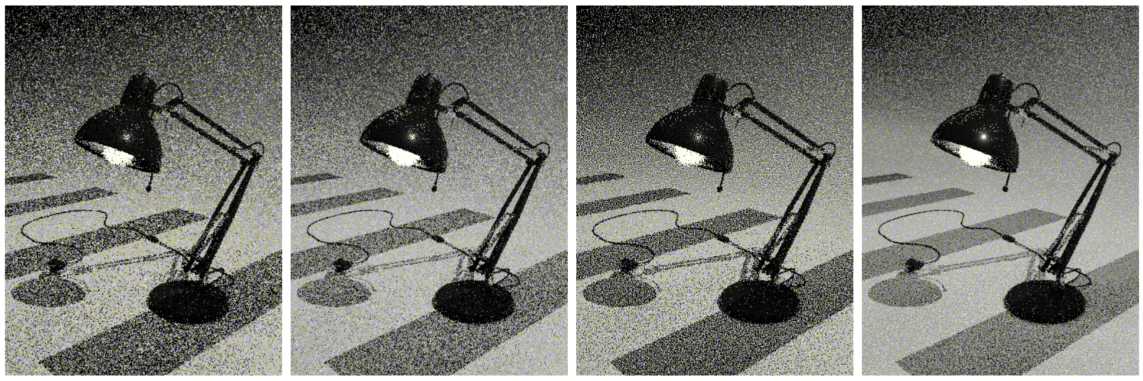


Figure 12: Total image Error



(a) 1 spp white (b) 2 spp white (c) 1 spp blue (d) 2 spp blue

Figure 13: Types of images from the 'Little Lamp' scene

sample density. Both to test whether one-sample blue noise indeed outperformed white noise with more samples, and to be able to provide a reference point to determine the scale of the difference between blue and white noise, we added higher-sample images to the experiment.
With four image types, comparing images one-by-one would require many comparisons to be made. One ranking of four images practically provides six comparisons due to transitivity. (If image A is ranked lower than image B, and image B is ranked lower than image C, then image A must also be ranked lower than image C). So, ranking makes it easier to compare four image types at once.

Second, it forces participants to make a distinction between images. If we asked the participants to rate the images, for example on a score from one to five, there would be a large chance that many images would get a low score because none of the images are 'good'. If scoring images the same is not an option, participants will be more motivated to find small differences.

Third, ranking is a puzzle-like task which makes the experiment more fun for participants, and happy participants provide better data (Ferwerda, 2008).

3.4.3 Task Implementation

We created the survey in Qualtrics XM [1]. This software is designed for creating surveys but it did not have all the specific functionalities we needed. Luckily, this was possible with a combination of Javascript and CSS.

The most important modification to be done was to scale the images correctly. Because this research concerns the higher frequencies of noise in images, it is very important that those frequencies are displayed correctly. This means that no blurring or interpolation must be applied on the images
To make sure that the images were displayed correctly, the pixels of the original images have to align perfectly with the actual displayed pixels on the screen. If they are not aligned, then there will be aliasing. In browsers, images are always being resized to fit the screen size nicely. By changing the CSS code in the survey, it was possible to force the images to keep their specific resolution and it was possible to disable any blurring. We did this by defining the image as a background-image of the element, because background images could be exempt from scaling. The code for an image in the survey is shown in Listing 1.
However, this did not fully fix the problem because some devices, such as laptops, also perform their own scaling that is separate from the browser's scaling. The browser is generally not aware of this scaling so this is not fixable by using CSS. Also, if anyone were to zoom in or out while filling in the survey, the scaling would also be changed. To remedy this, we had to use JavaScript. Qualtrics has the option to add JavaScript code to questions to enable special functionality. With

[1] https://www.qualtrics.com/

JavaScript, we were able to find the scaling ratio from ‘browser pixel’ to ‘screen pixel’ and scale the images accordingly whenever the survey got started. It also made sure the re-scale the images if a user zoomed in or out. The code for scaling the images is displayed in Listing 2. This code is called when the survey starts and whenever the window resizes.

```
<div class="stimimage"
style="
    width: 384px;
    height: 512px;
    max-width: 384px;
    background-size: auto 100%;
    background-repeat: no-repeat;
    background-position: center;
    background-image: url(
        [image_url]);
    image-rendering: pixelated;
    ">
</div>
```

Listing 1: HTML/CSS example of an image entry

By forcing the images to always be true to their resolution, the survey did become a bit less accessible. The images would not get scaled, so on low-resolution screens that means that they become very large compared to the screen. The images are also always shown side-by-side so the survey would also not work well on vertical screens, like mobile phone screens. There was not a good way to show the images at the correct size and the right layout on all devices without the participant having to scroll, so the solution we decided is to inform participants that the survey only works on computer and laptop screens. The resolution of each participant’s browser window is also collected by the survey program so that we can recognize any entries that were completed on a vertical screen.

```
var stimimages = document.getElementsByClassName('stimimage');
for (let item of stimimages) {
    item.style.width = (384/window.devicePixelRatio) + "px";
    item.style.height = (512/window.devicePixelRatio) + "px";
}
```

Listing 2: JavaScript example

While Qualtrics does have drag and drop functionality, it only supports vertical dragging. For this experiment we strongly preferred to align the images horizontally, so we could fit four images on a screen neatly. This required some additional modifications to the CSS code. This code is very specific to the implementation of Qualtrics and consists of a lot of small tweaks so the exact code is not included in this thesis.

3.4.4 Pilot

Before the final survey was released, we conducted a pilot experiment to test whether this task would work. This experiment consisted of the same images and was sent to ten people. The results from this pilot suggested that there indeed was a visible difference between the different image types, but that one-sample blue noise did not outperform two-sample white noise, as is shown in Figure 14. Because of this, we did not include three-sample images into the task. If the one-sample blue noise had been rated higher than the two-sample white noise, it would have been useful to add images with more samples so we could establish an upper bound.

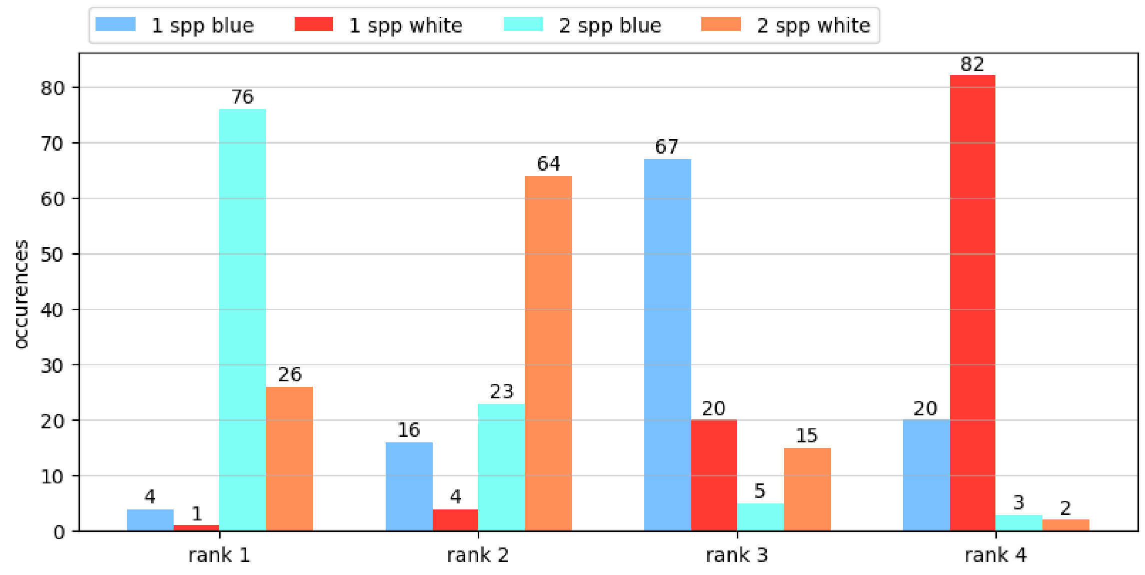


Figure 14: Rank distribution results from the pilot

3.4.5 Questionnaire

Prior to starting the task, participants were asked a couple of questions to gain data about our demographic target. The participants were asked about:

- Their age
- Whether they have normal or corrected to normal vision
- How much time they spend looking at screens.
- On what kind of screen they are seeing the survey.

The age of the participants is interesting to us for two reasons. The first reason is that we want to try to get responses from more groups than just students. Due to this being a university thesis, most responses will likely be from students or young people. By asking for the participant's age, we can get a rough idea of how diverse the reached audience was. The other reason to ask for age is because it allows us to research whether age has any influence on the results of this experiment. Age has been shown to have an effect on eyesight in general and, more specifically, on contrast sensitivity (Blackwell & Blackwell, 1971).

It is also important to know whether participants have normal or corrected to normal vision. If a person's vision is not normal and not corrected to normal, their spatial acuity (corresponding to the spatial frequency axis of the contrast sensitivity function) is lower than average.

The amount of time spent looking at screens is not likely to be an important factor, but it might have an effect. Spending a lot of time looking at screens might train the eyes to see high-frequency content better because screens often contain high-frequency content.

The kind of screen that the participant is looking at is important to estimate the size of the screen and distance to the screen. While we have the resolution data of the screen, that gives little information about the size of the screen. We do not want to ask participants to measure the size of their screen as that will probably be too big of a hurdle to do the survey. From the type of screen (laptop/pc monitor/tablet) we can find an average size of the screens and also estimate an average distance that a person would have to the screens. From that it becomes a little easier to estimate what size the pixels would be in the participant's field of view. This is further discussed in section 4.2.

3.4.6 Participants

We distributed the survey to participants through several channels:

- Friends and family.
- Student association BITON.
- The Utrecht University Visualisation and Graphics group.
- The Utrecht University Call for Participants Teams channel.
- Reddit group SampleSize[2].
- Through SurveySwap[3], a website where people can complete surveys in exchange for other people filling in their survey.

Similar vision experiments have around 19 to 40 participants (Johnson and Fairchild, 2005; Calabria and Fairchild, 2003; Hunt and Sera, 1978). Since this experiment is not in-person and there is the option for participants to leave parts unanswered, we wanted to have a larger amount of participants to ensure there was enough usable data. The goal was to reach at least 25 participants, but preferably around 50.

4 Results

The survey was completed by 56 participants. Figure 15 shows the demographic distribution. Most responses were from younger people. Only one respondent did not have normal or corrected to normal vision.

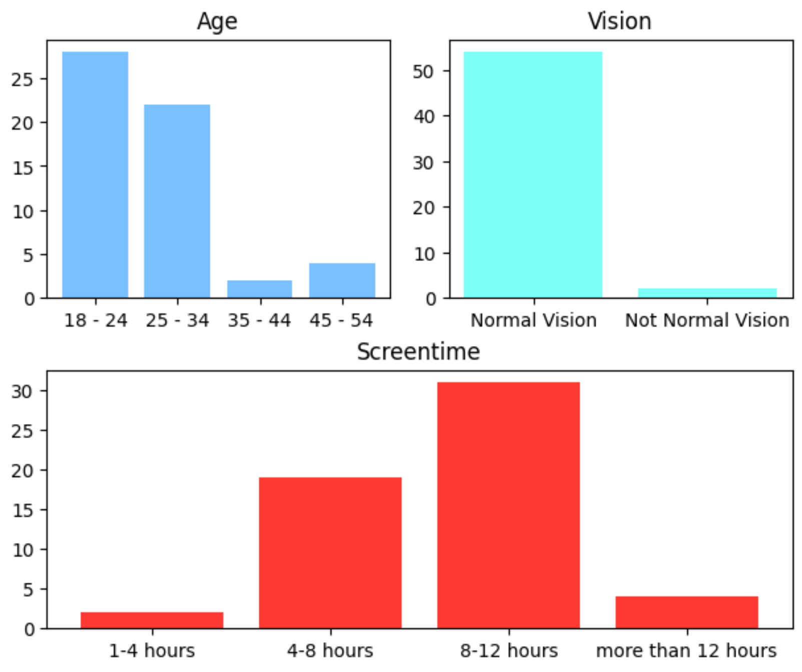


Figure 15: Demographic distributions

[2]https://reddit.com/r/SampleSize/
[3]https://surveyswap.io/

4.1 Analysis of the stimulus images

In order to check the validity of our results it is important to analyse the noise in the used stimulus images. We have to make sure that the renderer did correctly generate images with blue and white noise. To do this we have generated frequency spectra of the images and compared them to images that we know are blue and white noise.

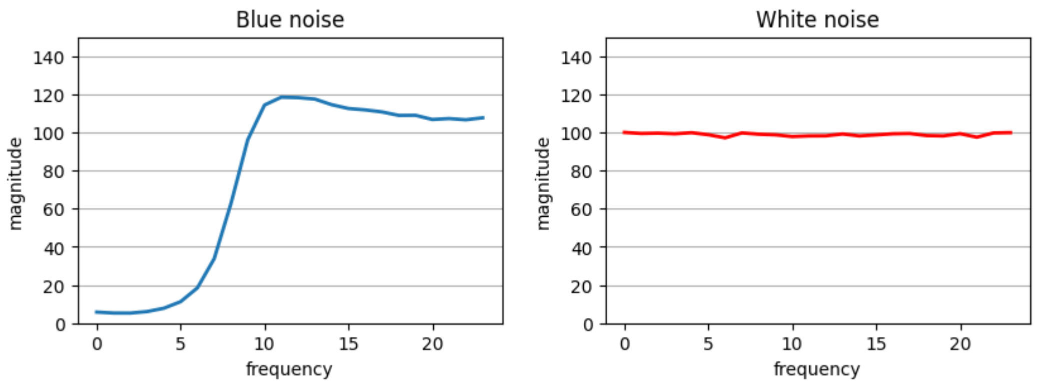


Figure 16: Frequency spectra of blue and white noise

For the reference blue noise and white noise, we used two online noise generators (Peters, 2016; Robson, 2021). To check these noise images, we analyse the frequencies. Shown in Figure 16 are the frequency spectra of the blue and white noise reference images. The frequency axis here is not scaled to represent actual frequency numbers, but corresponds to frequency bins numbered from 0 to 25, with 0 meaning the lowest frequencies possible in the image and 25 the highest frequencies possible in the image (Nyquist frequency).
You can clearly see the expected spectrum of the blue noise: small amount of low-frequency details, a ‘bump’ in the middle frequencies and then a slightly lower plateau on the higher frequencies. The white noise reference image also behaves as expected: each frequency is present in equal amounts.

We then did the same for each of the scenes. The frequency spectra for the 1-sample per pixel blue noise and white noise images were calculated and plotted together (Figure 17). This also includes a new ‘lampempty’ scene which is the lamp scene with the lamp object removed. This is used to gauge how well the renderer generates blue or white noise in a plain image. The image generated in this scene should not have any variation in it. This means that any high frequency details in the renders are noise. In Figure 17 you can see that in most scenes, the difference between the ‘zsobol’ and ‘independent’ sampler is quite small when compared to pure blue versus white noise. This is because the actual details present in the scene also show up in the frequency spectrum. There are some graphs which show a very clear difference between the ‘zsobol’ and ‘independent’ samplers like the *dining-room* and *lamp* scenes. The *staircase* and *living-room* however show almost no difference in frequency spectrum between the two samplers.

Not every scene has the same difference between the zsobol and independent renderers. In some scenes the rendering error gets distributed as blue noise more effectively than in others. It is interesting to see that this variation is also reflected in the participants’ ranking, as is discussed in section 4.4.2.

We determine a metric to measure the ‘blueness’ of the noise in the rendered images, so we can compare them more easily. To do this, we find the difference between the blue and white noise spectra of each scene. Shown in Figure 18 are the frequencies of the blue noise image minus the frequencies of the white noise image for the 1 sample-per-pixel renders (the blue line). We chose to use this as comparison data because the noise levels are different in each scene and using the difference between the blue and white noise removes that from the comparison. We are not

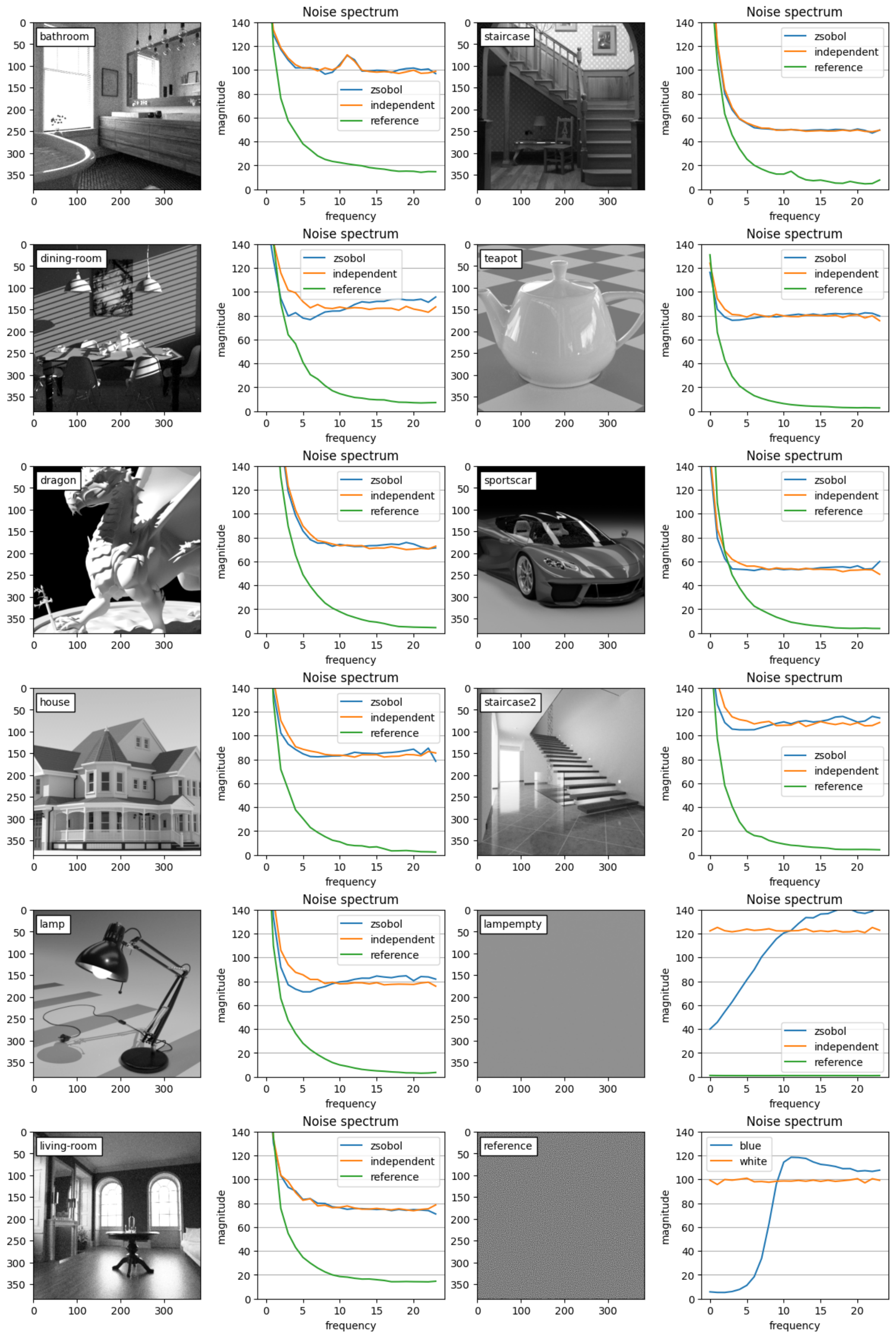


Figure 17: Noise spectra of the scenes

interested in absolute noise levels of the rendered scene; we are interested in how the noise frequencies are distributed for the blue noise compared to the white noise. The reference blue noise minus reference white noise spectrum is plotted as the orange line and is the same in each plot. This reference shows what the most ‘blue’ distribution that is possible in an image of this size looks like.

We can now look which graphs look the most like the reference line. That tells us in which scenes the difference between blue and white noise images is most equal to the difference between the reference blue and white noise. To find the similarity, we calculate the distance between the graphs using ‘cityblock distance’ or Manhattan distance which is just the sum of the vertical distances at each point of the x-axis. The distance this gives is also noted in the plot. The *lampempty* plot has the smallest distance which is to be expected as that scene has the least details and thus is mostly noise. The *staircase* and *living-room* have the highest distance which corresponds to what we saw in Figure 17: those are the scenes where there is little difference between the two renderers visible in the frequency spectrum.

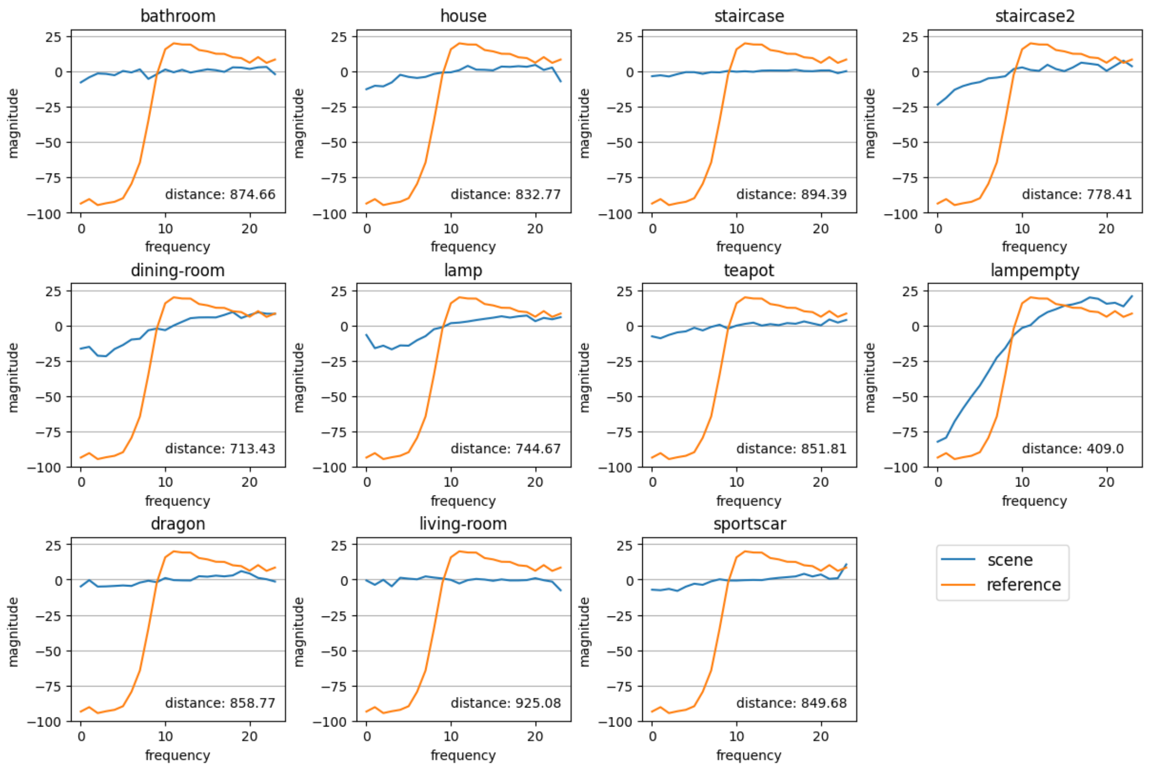


Figure 18: Noise spectrum differences

4.2 Screen resolutions

The screen size of participants was recorded in order to estimate the average size of a pixel on the participant’s screen. As can be seen in Figure 15, there are three outliers with a very small resolution scale (390x844 pixels or less). These aspect ratios also signify that these are portrait mode screens, and they are common viewport sizes for mobile phones. Even though two of the entries are labeled as tablet screens and one is labeled as a computer monitor, it seems very likely that these surveys were completed on mobile phones.
It is important to note that the resolution that is recorded in the survey is not the actual resolution that is displayed on the screen. It is the resolution that the web browser viewport thinks it is. Modern mobile phones can have higher resolutions than computer screens, but the viewport

resolution would still be low. This makes text and images automatically show up larger on mobile phone screens when there is not a version of the website made specifically for mobile. If phones did not report this ‘fake’ resolution to the web browser, websites would have the same proportions as on a computer display but just scaled down to phone size, making it impossible to read anything. Many laptops do the same thing with a more subtle difference, making everything for example 25% larger to make up for screens that are usually smaller than desktop monitors.

Because of this, the images in the experiment could have been properly displayed (full resolution) on a mobile phone screen, but they would have been very small.
An interesting thing to note is that the results from these three entries are not very different from the other results (Figure 19). The distribution of ranks is comparable to the distribution of ranks in all other entries. While completing the experiment on a mobile phone was discouraged because it would be hard to navigate, it seems that some participants did manage. The data from these entries are not outliers and therefore not necessary to remove. However, when determining the average pixel size, these entries are definitely outliers and should be excluded.

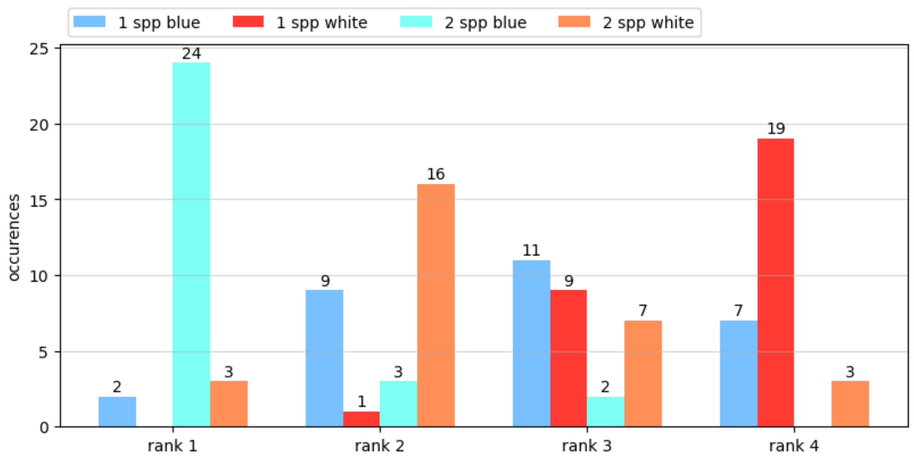


Figure 19: Rank distribution for outlier resolutions

As for the other resolutions, these are all within the acceptable range. Again noting that these are not necessarily the actual screen resolutions, the lowest one of 1280x720 pixels corresponds to using a 150% zoom level on a 1920x1080 screen. The other frequently occurring ‘weird’ resolution is 1536x864, which corresponds to using a 125% zoom level on a 1920x1080 screen. We have tested and confirmed that the survey still displays correctly in these circumstances.
It is possible that a survey was completed on a 1280x720 resolution screen. It is a common screen resolution for older monitors, but quite rare for modern monitors. In the case that a screen with this resolution, the participant would only be able to see three images at a time and would have had to scroll sideways to complete the ranking task. Doing so would make the task more inconvenient but not impossible, so we can also accept this screen resolution.

The goal of collecting the resolution data was to use it in combination with worldwide averages for screen sizes and user-to-screen distances to estimate the size of the stimulus images for participants. Unfortunately, such statistics were not available and we had to make some assumptions to perform this part of the research.

The first two assumptions are about screen sizes. A user on *strawpoll.com* has asked other users about their favourite laptop screen and monitor sizes. For laptop screens, 15.6 inches got the most votes with $\sim 34\%$ of the votes (Krambs, 2024a). For monitors, 24 inches got the most votes with $\sim 32\%$ of the votes (Krambs, 2024b). This data is self-reported and has a low sample size, but it does give us a general idea of screen sizes.

For average screen distances there are no measurements available, but there are recommended distances. As stated on the website of the College of Optometrists (2024), the recommended screen-to-eye distance is between 40 and 76 centimetres. For our estimate we pick the average: 58 centimetres.

We want to know whether the csf (contrast sensitivity function) can be used to explain why blue noise is preferable to white noise. Does the distribution of error as blue noise push the noise into such high frequencies that it is less visible to the human eye? To answer this question, we should find how the csf translates to the frequency spectra we generated of the used images.

The csf is a function of *contrast* against *cycles per degree*. Cycles per degree is the amount of cycles that are visible in one degree in the field of vision. The frequency spectra are arranged in frequency bins with no direct connection to cycles per degree. We can use the screen resolution measurements and the other estimates to find an average pixels-per-degree for the participants. With that, we can find the maximum cycles-per-degree that the screens can show.

With standard screen aspect ratios (16:9), a 15.6 inch diameter screen is 345 millimetres wide. A 24 inch diameter screen is 531 millimetres wide. For each screen we calculate the visual angle that a single pixel takes up:

$$visualangle = \arctan(\frac{screenwidth}{distance}) * \frac{1}{resolutionwidth}$$

And inverting it gives us the amount of pixels per degree:

$$pixelsperdegree = 1/visualangle$$

The estimated average pixels per degree of all screens is 48 with a standard deviation of 6.5.

To anchor the frequency bins to a specific number of cycles per degree, we have generated images with 2, 3, 4, 8, and 20 pixels per cycle. These images are one-directional sinusoidal gratings. When applying a Fourier transform on these images and sorting them into the frequency bins, we find where on the frequency spectrum those frequencies fall. Because we know the amount of pixels per degree of each screen, we can now express those frequency bins in approximate cycles per degree:

pixels per cycle	2	3	4	8	20
cycles per degree	24	16	12	6	2.4

We can plot these points on the frequency spectrum of the *lampempty* scene to get an idea of how these frequencies relate (Figure 20). The cycles per degree have a linear relationship with the frequency bins which allows us to easily add cpd as a secondary horizontal axis. We can compare this to the contrast sensitivity function (Figure 21), note that the latter is depicted in logarithmic scale on both axes. In Figure 21 we see that the sensitivity is highest between a spatial frequency of one to five cycles per degree (a sensitivity of around 300). At 10 cpd the sensitivity has dropped to around 100. At 20 cpd the sensitivity is only 20. At 30 cpd, the contrast sensitivity is somewhere between two and three.

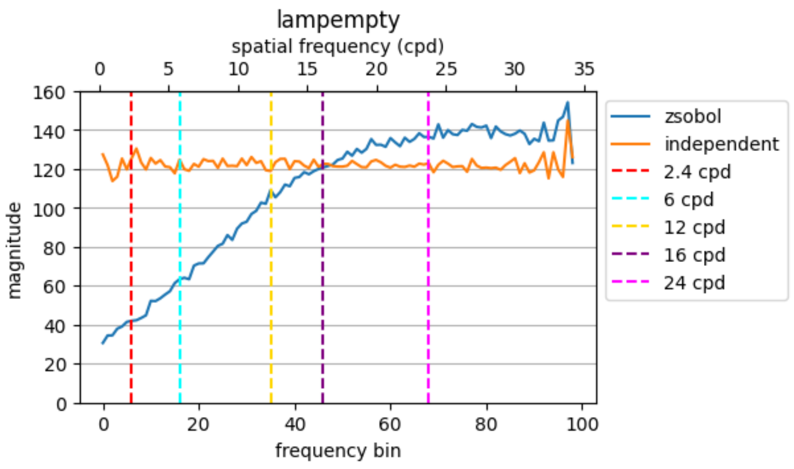


Figure 20: frequency spectrum with spatial frequency axis

Looking at Figure 20, we see that the 1-5 cpd area with the highest sensitivity is actually only quite a small part of this plot. We also see that there is a lot less noise in that area for the *zsobol* sampler than for the *independent* sampler. The lines intersect around 16 cpd which is definitely still within visible range but the sensitivity is only one tenth of the maximum. The maximum spatial frequency of the images is below 35 cycles per degree. This is still within visible range; humans are able to detect spatial frequencies of up to about 60 cpd (National Research Council, 1985). However, the sensitivity at 35 cpd is already less than one hundredth of the maximum.

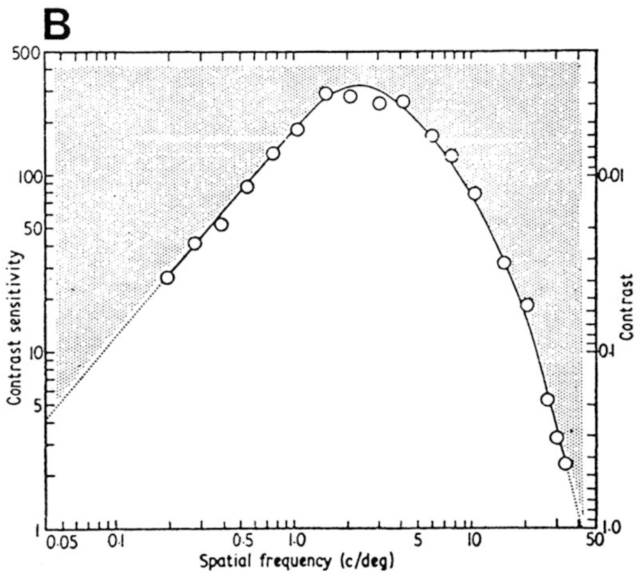


Figure 21: Photopic contrast sensitivity function of the human visual system for sinusoidal gratings (source: National Research Council, 1985)

In Figure 22 we see the frequency spectra of the *dining-room* scene with the overlaid cpd anchors. This example shows that most of the detail of a scene is in the frequency range of one to ten cycles per degree.
In short, blue noise has less energy than white noise in the frequency range that the human eye is most sensitive to. It has more energy in the higher frequencies which are less noticeable to the human eye.

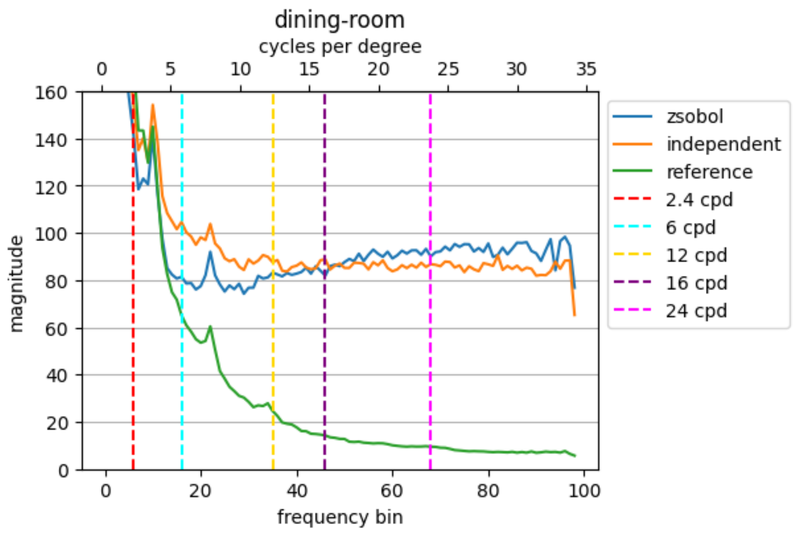


Figure 22: frequency spectrum with spatial frequency axis for the dining-room scene

4.3 Ranking Data

The main data collected from the experiment is a ranking of each noise type per scene and participant. In total, 56 participants completed the survey. This could give us a potential of 560 rankings, but not all participants ranked each scene. The initial order in which participants saw the noise types was randomised to minimise bias. A downside is that the survey software classified questions as unanswered in the case that a participant was satisfied with the initial ranking and did not move any images around. The other reason why a ranking would be left unanswered would be because a participant could not identify any differences between the images and decided to skip the task. In total, 30 rankings were left unanswered, leaving us with 530 rankings.

The data is summarised by noise type in Table 1. This shows the number of times each noise type was chosen as each rank. The same data is displayed in Figure 23 as a bar graph. In this data, you can see that each noise type has a clear 'peak' at a specific rank. These peaks contain more than half of all occurrences.

	rank 4	rank 3	rank 2	rank 1
1 spp white	326	128	41	35
1 spp blue	137	278	76	39
2 spp white	36	85	298	111
2 spp blue	31	39	115	345

Table 1: Rank distribution per noise type

4.3.1 Limitations of rank data

The collected data are ranks. While rank data is similar to numerical data, they cannot be used in the same way. Ranks are ordinal data, meaning they have a certain order. It is possible to compare two ranks and say one is higher or lower than the other, and it is possible to sort them. However, it is not possible to measure the distance between ranks (Cunningham & Wallraven, 2011, p.223). While for normal numbers it is correct to state that the difference between 1 and 2 is equal to the difference between 2 and 3, it is not correct to say that the difference between rank 1 and rank 2 is equal to the difference between rank 2 and rank 3. Consider a ranking of images, like in the experiment. We have no way of knowing whether a participant thought there was a huge

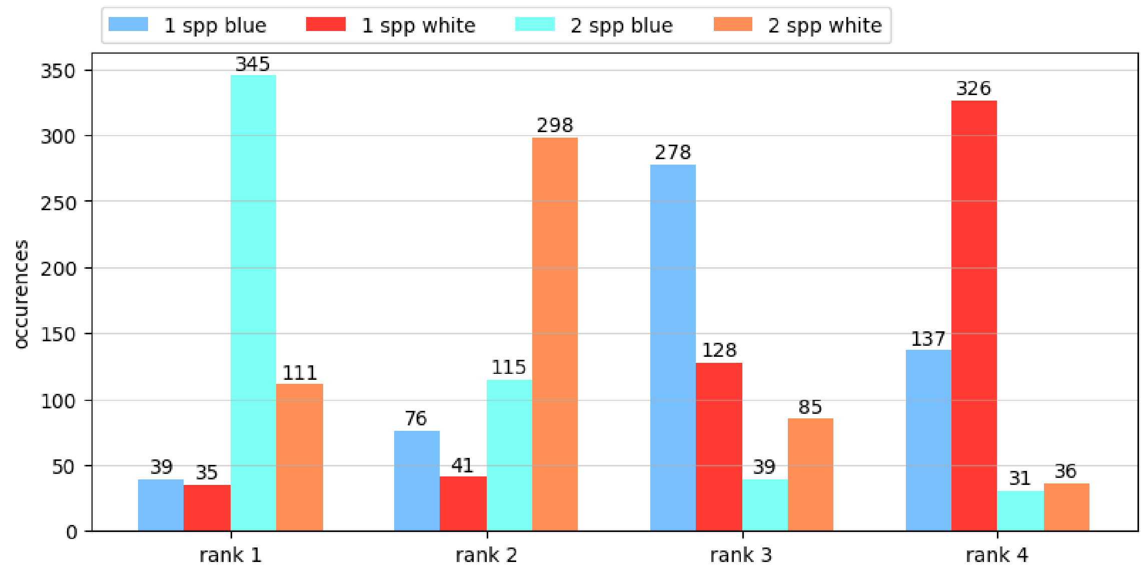


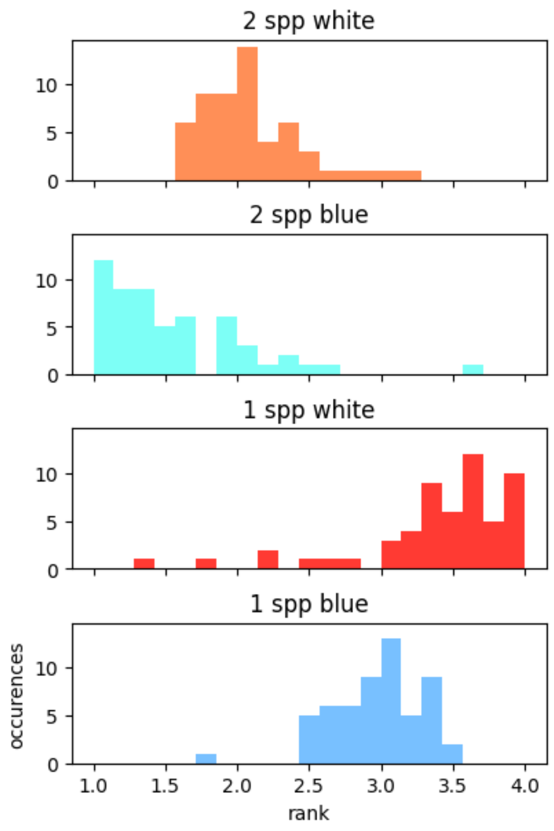
Figure 23: Rank distribution per noise type

difference between images ranked third and fourth, or that they barely saw a difference. This means that calculating the average of rank data is not statistically correct, and the resulting average is invalid. This results in limitations to the types of statistical tests that can be performed on the data. This is discussed in section 4.4.
While averaging is not a statistically sound way to draw conclusions, it can still help our understanding of the data as long as we keep the limitations in mind. This is why some graphs shown later in the this section do apply averaging over ranks.

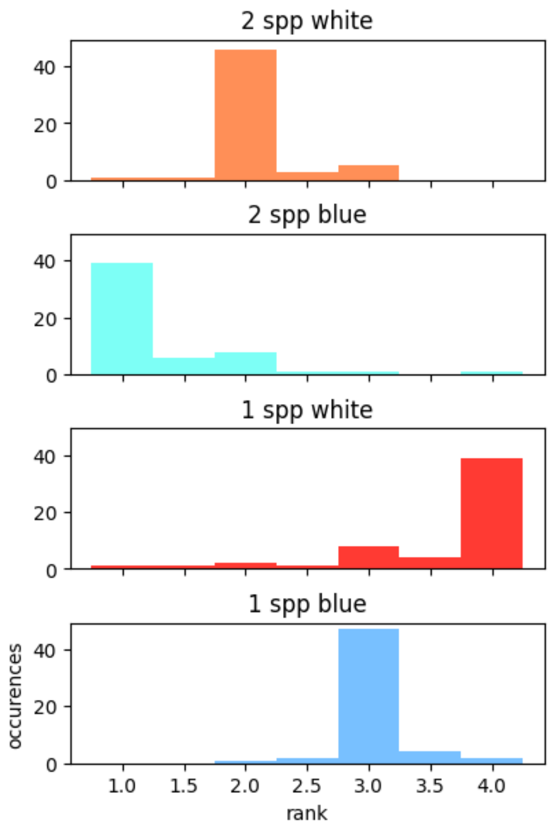
4.3.2 Data Visualisation

Taking the mean of ranking data does not provide meaningful numbers, but it does give a better idea of how the ranking results were distributed. If we look at a single participant and calculate the average rank that they have given to a specific noise type in all scenes, that number gives us an idea of what that participant thought of that specific noise type. In Figure 24a you can see the distribution of the average rank that participants ranked a noise type.

A more statistically correct representation would be to show the median of the ranks per scene. Finding the median of a data series does not involve calculations so it can be done for ordinal data. This is not totally true when the median is in between two numbers, in which case the mean of those numbers is calculated. The halfway point between two ordinal data points is a meaningful indication though, as that indicates that both points are the middle of the data. An important thing to note here is that the middle point does not mean that it is equally far away from both points. For example, the median rank 1.5 is not exactly in the middle of 1 and 2 but can be anywhere in between.
Taking the median of ranks per participant for each noise type and plotting that as a histogram gives us Figure 24b. This shows that many participants attribute similar ranks to the noise types.



(a) Mean ranks



(b) Median ranks

Figure 24: Average ranks per participant

4.4 Statistical tests

In order to test whether the difference in the distribution of ranks between the noise types is statistically significant, we can apply several statistical tests.
The data we are testing is ordinal data, which greatly limits the number of applicable statistical tests. We have identified four tests which are applicable to this data: the Sign test and the Mann-Whitney U test only test two categories against each other. The Kruskal-Wallis H-test and Friedman test analyse all categories at the same time (Corder & Foreman, 2014). We are interested in comparing two categories, as we can already predict that not all categories are from the same distribution because *2 spp blue* and *1 spp white* have very different averages. The two-category tests will be able to tell us more detailed information on categories that are closer together.

The sign test and Mann-Whitney U test are both tests that work for non-parametric data. The sign test is used to test between pairs of dependent samples while the Mann-Whitney U test is used to test between independent samples. Rankings for the different noise categories are dependent samples because they are always ranked by the same participant. That is why the Sign test is best suited for this data.
The Mann-Whitney U test is also not a very good fit for the data in this experiment. The basic idea of the test is ordering all data points of both categories in one list and seeing if the points from each category are well mixed or clustered at either end of the list. Because there are only four ranks, this means that ordered list will consist mostly of tied ranks which cannot be internally ordered. Though the Mann-Whitney U test does have a way of dealing with some tied ranks, it seems that our data does not fit the intended use for this test.

4.4.1 Sign test

The sign test is used to test for consistent differences between pairs of observations. It takes pairs of data that can be smaller than, equal to, or larger than each other. It is a non-parametric test which requires very few assumptions about the distributions of the testing data. Given a set of comparisons, it can test whether the distribution of these comparisons is fitting for the hypothesised distribution.

We apply this test to *1 spp white* with *1 spp blue*, *2 spp white* with *2 spp blue*, and *2 spp white* with *1 spp blue*.

We show the calculation for the comparison of 1 sample-per-pixel white noise with 1 sample-per-pixel blue noise: In our data we found that, out of 530 comparisons, blue noise was ranked better than white noise in 360 cases and white noise was ranked better than blue noise in 170 cases. We do a one-sided test, so we will test whether blue noise is scored better than white noise more often than expected. The null hypothesis is that the distributions are equal, so a 50% chance that either noise type comes out on top. In this scenario we would expect blue noise to beat white noise in around 265 cases. The alternative hypothesis is that blue noise is more visually preferable than white noise and thus have a higher chance chance of being ranked better.

Then we ask what is the probability of 360 out of 530 comparisons to be positive for blue noise if the null hypothesis were true? This is the same probability as throwing up a coin 530 times and it coming up as heads 360 times which can be calculated with the binomial test. In a sample of size n where the expected chance of success is p, the probability of finding k successes is:

$$\binom{n}{k} p^k (1-p)^{n-k}$$

In order to find the p-value for this test, the binomial test considers the probability of seeing an outcome that is equal or more extreme. This means we sum the probabilities from k to n. Our p-value becomes:

$$\sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

In the case of $n = 530$, $k = 360$ and $p = 0.5$, this gives us a p-value of $5.22556e^{-17}$ which is smaller than the required p-value of 0.05 and thus we can reject the null hypothesis that 1 spp blue noise only has a 50% chance of beating 1 spp white noise.

We can do the same calculation for 2 sample-per-pixel white and blue noise. In that case n is again 530 and p is 0.5. In this case there are 387 cases where blue noise scored higher than white noise, so $k = 387$. Using the same equation, we find that $p = 2.83561e^{-27}$ meaning we can reject the null hypothesis that 2 spp blue noise and 2 spp white noise have an equal chance of getting ranked higher than the other.

Since it looks like 2-sample white noise consistently ranks higher than 1-sample blue noise, we can also test whether that is significant. 2-sample white noise ranks higher than 1-sample blue noise in 418 of the 530 cases. We can already guess that the probability of this is even lower because the number of successes is even higher than the previous test. The p-value here is $8.28229e^{-43}$ so we reject the null hypothesis that 2-sample white noise gets ranked equally to 1-sample blue noises.

We can conclude that the participants clearly show preferences between the noise types. We can accept the hypotheses that:

- 2 sample-per pixel blue noise is ranked higher than 2 sample-per-pixel white noise, more often than can be explained by chance
- 2 sample-per pixel white noise is ranked higher than 1 sample-per-pixel blue noise, more often than can be explained by chance

- 1 sample-per pixel blue noise is ranked higher than 1 sample-per-pixel white noise, more often than can be explained by chance

4.4.2 Significance per scene

We can find the statistical significance of the ranking differences for each scene, also using the sign test. Table 2 shows the p-values for all scenes for the hypothesis that 1 spp blue noise ranks higher than 1 spp white noise. We see that the results are most significant for the *lamp* and *house* scenes and are not significant for *staircase* and *living-room*. This is interesting because *staircase* and *living-room* were also the two scenes where the noise difference between blue and white noise was the smallest (discussed in section 4.1). This indicates a possible relation between difference in noise and the difference between perceived pleasantness. See Figure 25 for a visual representation.

Scene	**p**	**significant**
Bathroom:	0.0033	Yes
Dining_room:	0.0006	Yes
Dragon:	0.0007	Yes
House:	0.0002	Yes
Lamp:	1.9457e-10	Yes
Living_room:	0.5551	No
Sportscar:	0.0111	Yes
Staircase:	0.3359	No
Staircase2:	0.0267	Yes
Teapot:	0.0380	Yes

Table 2: Significance of results for different scenes

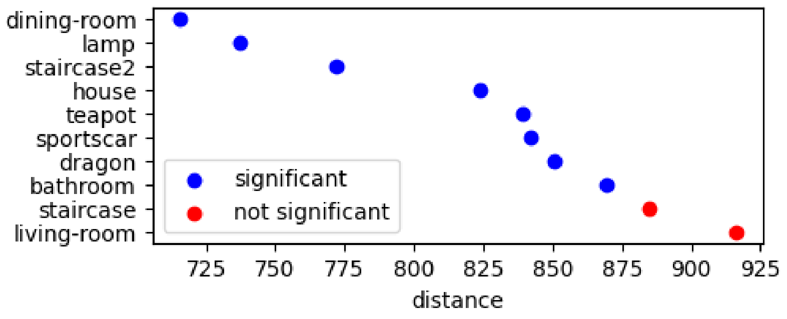


Figure 25: Relation between distance measure and signifance

5 Conclusion

5.1 Findings

The research question we set out to answer is the following:
Is a rendered image with the error distributed as blue noise visually preferable over a rendered image with the same amount of error distributed as white noise?

We can conclude that, indeed, blue noise is visually preferable to white noise according to human observers. We can also conclude that the visual preference of blue noise over white noise does not outweigh the visual improvement of rendering an image with two samples per pixel instead of one. With the decorrelated sampler implementation used in this thesis, the computation cost of using blue noise instead of white noise is only a fraction of the cost of taking another sample (Ahmed & Wonka, 2020). So if the goal is improving visual pleasantness, blue noise is a computationally cheap method to achieve that. Added to that, usage of blue noise instead of white noise already reduces the objective rendering error when taking as little as two samples (Figure 12).

We also find that the visual preference for blue noise versus white noise is not equal in all of the scenes. In two cases, blue noise was not ranked significantly better than white noise, but it was also not ranked worse. When looking at the frequency spectra of these scenes we see that there is barely any difference between the blue noise and white noise images (Figure 18). This means that the sampler was not able to produce a blue noise distribution of rendering error as effectively as for other images. It is not directly clear why this is the case for these scenes. It could be connected to the complexity of rendering the scene or the amount of high-frequency content in the scene.

The contrast sensitivity function of human vision is a very likely explanation for the preference for blue noise. When image error is distributed as blue noise, it contains less low-frequency content and more high-frequency content. Because the human eye is more sensitive to low-frequency content, this results in the error being less visible overall. With the current average screen types and distance from eye to screen, the high frequency noise that remains in blue noise is still well within human perceptible range. In order for the blue noise to be out of human perceptible range, displays would need to have a resolution sixteen times as much as the average resolution while being the same size as current displays. However, because the important content of a render or any natural image is mostly contained within the lowest frequencies which we are most sensitive to, that content draws the attention more. The strong point of blue noise is thus that it contains minimal power in the frequencies that draw the most attention and therefore blue noise does not interfere much with the actual content of an image.

Two-sample white noise is found to be preferred over one-sample blue noise so it seems that whichever image has the smallest amount of visible noise is deemed most visually pleasant. Therefore it might not be fully correct to call blue noise more visually pleasant than white noise. Rather, it is less visible which means it is less distracting from the important content of an image. This property of being less obtrusive makes blue noise more visually pleasant than white noise when used in image rendering.

5.2 Limitations and Future Work

The participants of this research are primarily young people. Ideally, this research would have had a more diverse group of participants that reflects the demographics of society. This would make generalisations made based on the data more reliable.

In order to make stronger statements about the visibility of blue noise in reference to the contrast sensitivity function, it would have been useful to have more accurate information about participants' screen size, screen resolution, and distance to the screen. This is not possible in the online format of this experiment, but it would be feasible if the experiment is carried out in-person. Performing a similar experiment with varying levels of contrast, one would be able to test whether the contrast sensitivity function accurately describes the visibility of blue noise.

This experiment only tested two amounts of samples per pixel for both noise types. This let us to find that 1-sample blue noise is better than 1-sample white noise but not better than 2-sample white noise. It does tell us how many samples of white noise would be equal to one sample of blue noise. An experiment which tests more amounts of samples per pixel for both noise types against each other would allow researchers to find which levels of sample per pixel for both blue and white noise produce perceptually equally good images. This could be used as a benchmark for developers to understand how much perceptual visual improvement their rendering system would gain from implementing blue noise. It would make it possible to weigh perceptual improvement against computation costs in an quantitative way.

A contrast sensitivity function for temporal frequencies also exists. The human eye is most sensitive to frequencies around 5Hz and are able to see up to 30Hz (Granrath, 1981). There is research into temporal blue noise where the distribution is not only blue in screen-space but also for successive frames (Wolfe et al., 2022). Future research could be conducted to see how this might be used for video games or other media with a high amount of frames per second.

References

Ahmed, A. G., & Wonka, P. (2020). Screen-space blue-noise diffusion of monte carlo sampling error via hierarchical ordering of pixels. *ACM Transactions on Graphics (TOG)*, *39*(6), 1–15. https://doi.org/10.1145/3414685.3417881

Barten, P. G. (1989). The square root integral (sqri): A new metric to describe the effect of various display parameters on perceived image quality. In *Human vision, visual processing, and digital display* (Vol. 1077, pp. 73–82). https://doi.org/10.1117/12.952705

Blackwell, O. M., & Blackwell, H. R. (1971). Visual performance data for 156 normal observers of various ages. *Journal of the Illuminating Engineering Society*, *1*(1), 3–13. https://doi.org/10.1080/00994480.1971.10732194

Bolin, M. R., & Meyer, G. W. (1998). A perceptually based adaptive sampling algorithm. In *Proceedings of the 25th annual conference on computer graphics and interactive techniques* (pp. 299–309).

Calabria, A. J., & Fairchild, M. D. (2003). Perceived image contrast and observer preference i. the effects of lightness, chroma, and sharpness manipulations on contrast perception. *Journal of imaging Science and Technology*, *47*(6), 479–493.

Chizhov, V., Georgiev, I., Myszkowski, K., & Singh, G. (2022). Perceptual error optimization for monte carlo rendering. *ACM Transactions on Graphics (TOG)*, *41*(3), 1–17. https://doi.org/10.1145/3504002

College of Optometrists. (2024). Screen use. Web Page. Retrieved from https://lookafteryoureyes.org/eye-care/screen-use

Corder, G. W., & Foreman, D. I. (2014). *Nonparametric statistics: A step-by-step approach.* John Wiley & Sons.

Cunningham, D. W., & Wallraven, C. (2011). *Experimental design: From user studies to psychophysics.* CRC Press.

Daly, S. J. (1992). Visible differences predictor: An algorithm for the assessment of image fidelity. In *Human vision, visual processing, and digital display iii* (Vol. 1666, pp. 2–15). https://doi.org/10.1117/12.135952

De Goes, F., Breeden, K., Ostromoukhov, V., & Desbrun, M. (2012). Blue noise through optimal transport. *ACM Transactions on Graphics (TOG)*, *31*(6), 1–11. https://doi.org/10.1145/2366145.2366190

Ferwerda, J. A. (2008). Psychophysics 101: How to run perception experiments in computer graphics. Conference Paper. 10.1145/1401132.1401243

Field, D. J. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *Josa a*, *4*(12), 2379–2394. https://doi.org/10.1364/JOSAA.4.002379

Floyd, R. (1975). An adaptive algorithm for spatial grey scale. *SID digest, 1975.*

Georgiev, I., & Fajardo, M. (2016). Blue-noise dithered sampling. In *Acm siggraph 2016 talks* (pp. 1–1). https://doi.org/10.1145/2897839.2927430

Granrath, D. J. (1981). The role of human visual models in image processing. *Proceedings of the IEEE*, *69*(5), 552–561. https://doi.org/10.1109/PROC.1981.12024

Heitz, E., & Belcour, L. (2019). Distributing monte carlo errors as a blue noise in screen space by permuting pixel seeds between frames. In *Computer graphics forum* (Vol. 38, pp. 149–158). https://doi.org/10.1111/cgf.13778

Holladay, J. T. (1997). Proper method for calculating average visual acuity. *Journal of refractive surgery*, *13*(4), 388–391. https://doi.org/10.3928/1081-597X-19970701-1

Hunt, B., & Sera, G. (1978). Power-law stimulus-response models for measures of image quality in nonperformance environments. *IEEE Transactions on Systems, Man, and Cybernetics*, *8*(11), 781–791. https://doi.org/10.1109/TSMC.1978.4309865

Johnson, G. M., & Fairchild, M. D. (2005). The effect of opponent noise on image quality. In *Image quality and system performance ii* (Vol. 5668, pp. 82–89). https://doi.org/10.1117/12.587384

Kelly, D. (1977). Visual contrast sensitivity. *Optica Acta: International Journal of Optics*, *24*(2), 107–129. https://doi.org/10.1080/713819495

Krambs, G. (2024a). The most popular laptop screen size: Revealing user preferences. Web Page. Retrieved from https://strawpoll.com/most-popular-laptop-screen-size

Krambs, G. (2024b). The most popular monitor size: Revealing the top choice for display screens. Web Page. Retrieved from https://strawpoll.com/most-popular-monitor-size

Landau, H. (1967). Sampling, data transmission, and the nyquist rate. *Proceedings of the IEEE*, *55*(10), 1701–1706. https://doi.org/10.1109/PROC.1967.5962

Lubin, J. (1995). A visual discrimination model for imaging system design and evaluation. In *Vision models for target detection and recognition: In memory of arthur menendez* (pp. 245–283). https://doi.org/10.1142/9789812831200˙0010

Mantiuk, R., Daly, S. J., Myszkowski, K., & Seidel, H.-P. (2005). Predicting visible differences in high dynamic range images: Model and its calibration. In *Human vision and electronic imaging x* (Vol. 5666, pp. 204–214). https://doi.org/10.1117/12.586757

Mitchell, D. P. (1987). Generating antialiased images at low sampling densities. In *Proceedings of the 14th annual conference on computer graphics and interactive techniques* (pp. 65–72). https://doi.org/10.1145/37401.37410

Mitchell, D. P. (1991). Spectrally optimal sampling for distribution ray tracing. In *Proceedings of the 18th annual conference on computer graphics and interactive techniques* (pp. 157–164). https://doi.org/10.1145/122718.122736

Morton, G. M. (1966). A computer oriented geodetic data base and a new technique in file sequencing. *IBM*.

National Research Council. (1985). *Emergent techniques for assessment of visual performance.* National Academies Press (US).

NVIDIA. (2023). Rtx technology. Web Page. Date accessed: 20/06/2023. Retrieved from https://developer.nvidia.com/rtx/ray-tracing

Pappas, T. N., & Neuhoff, D. L. (1999). Least-squares model-based halftoning. *IEEE Transactions on image processing*, *8*(8), 1102–1116. https://doi.org/10.1117/12.135965

Peters, C. (2016). Free blue noise textures. Web Page. Retrieved from https://momentsingraphics.de/BlueNoise.html

Pharr, M. (2020). Pbrt-v4-scenes. Dataset. Retrieved from https://github.com/mmp/pbrt-v4-scenes

Pharr, M. (2021). Physically based rendering toolkit. Web Page. Retrieved from https://github.com/mmp/pbrt-v4

Pharr, M., Jakob, W., & Humphreys, G. (2023). *Physically based rendering: From theory to implementation.* MIT Press.

Ripley, B. D. (1977). Modelling spatial patterns. *Journal of the Royal Statistical Society: Series B (Methodological)*, *39*(2), 172–192. https://doi.org/10.1111/j.2517-6161.1977.tb01615.x

Robson. (2021). White noise image generator. Web Page. Retrieved from https://robson.plus/white-noise-image-generator/

Sullivan, J. R., Ray, L. A., & Miller, R. (1991). Design of minimum visual modulation halftone patterns. *IEEE Transactions on Systems, Man, and Cybernetics*, *21*(1), 33–38. https://doi.org/10.1109/21.101134

Ulichney, R. A. (1987). *Digital halftoning.* MIT press.

Ulichney, R. A. (1988). Dithering with blue noise. *Proceedings of the IEEE*, *76*(1), 56–79. https://doi.org/10.1109/5.3288

Voss, R. F., & Clarke, J. (1978). ”1/f noise”in music: Music from 1/f noise. *The Journal of the Acoustical Society of America*, *63*(1), 258–263. https://doi.org/10.1121/1.381721

Wolfe, A., Morrical, N., Akenine-Möller, T., & Ramamoorthi, R. (2022). Spatiotemporal blue noise masks. In *Eurographics symposium on rendering.* https://doi.org/10.48550/arXiv.2112.09629

Yellott, J. I. (1983). Spectral consequences of photoreceptor sampling in the rhesus retina. *Science*, *221*(4608), 382–385. https://doi.org/10.1126/science.6867716