Dither

Goal: To obtain natural looking images without false contours

Problem: Using quantization with too few quantization levels, or histogram equalization in areas where the mapping function $g(x)$ is relatively steep, can lead to too big distances between neighboring quantization intensity levels. This can lead to artificial or false contours in the image.

Example:

Approach: The cause of the false contours can be seen in the histogram, which then has big gaps between neighboring intensity lines:
Discrete intensity lines are created by a quantizer:

The discrete intensity lines in the histogram are caused by luminance jumps of the quantizer (analog images have a continuous histogram).

In images, we usually have 8 bits per intensity value, which means only 256 distinct intensity values. Usually these 8 bits are sufficient to avoid false contours, but applying a mapping function $g(x)$ can increase the distance between some lines, and hence create those visible false contours.
What can we do to remedy this problem?

Answer: We need to insert additional intensity levels, to reduce the distance between the intensity levels. Simply applying another mapping function would not work, because it does not increase the number of intensity values. Hence we need a random function to generate additional intensity values. We simply add random numbers to each intensity, such that the gaps between intensity lines are sufficiently reduced. The range of the random number hence has to be adapted to the intensity gap size at each intensity level. This increases the quantization error, but is still more visually pleasing. To avoid this increase in quantization error, one could also use pseudo-random numbers, add them before quantization and subtract them after quantization, as in the following picture:

![Diagram of quantization noise decorrelation](Picture from: Jae. S. Lim, Two-Dimensional Signal and Image Processing, p.620)
A mathematical analysis:

Dither only on the decoder side. What is the mean squared error? We have a quantizer, which we can write as \( f(x) = x + \text{eps} \), where eps is a random number between -0.5 and 0.5, uniformly distributed (for a quantization interval of size 1).

The reconstructed value is then \( y(x) = f(x) = x + \text{eps} \).

The mean squared error is

\[
E((y(x) - x)^2) = E((x + \text{eps} - x)^2) = E(\text{eps}^2) = \frac{1}{12}
\]

Add dither \( d \) on the decoder side, where \( d \) is a random number in a range of \(-r/2\) to \(+r/2\):

\( Y(x) = x + \text{eps} + d \)

The mean squared error is then:

\[
E((y(x) - x)^2) = E((\text{eps} + d)^2) \\
= E(\text{eps}^2 + d^2 + 2\text{eps} \cdot d)
\]

We assume that eps and \( d \) are zero mean and independent. The latter means \( E(\text{eps} \cdot d) = 0 \). Hence

\[
E((y(x) - x)^2) = E(\text{eps}^2 + d^2 + 2e) \\
= E(\text{eps}^2) + E(d^2) = \frac{1}{12} + \frac{r^2}{12}
\]

Which means our reconstruction error increased.
Now imagine a system, where we first subtract the dither \( d \) before we quantize it, and the add the same dither \( d \) to the quantized value. The resulting quantization function is:

\[
Y(x) = x - d + \text{eps} + d.
\]

Hence the mean squared quantization error is:

\[
E((y - x)^2) = E(\text{eps}^2) = \frac{1}{12}
\]

Which is the same as before!

Observe: The first step of subtracting dither in the encoder is only possible if we also have designed the encoder. If we already have a given image, this is of course not possible. In this case we only have the second step, the addition of dither in the decoder.

To dither a histogram equalized image, we first need to take a look at the distance between neighboring intensity values in the histogram, and then add dither \( d \) of a size that is adapted to this distance, such that it sufficiently decreases this gap to avoid the false contours (meaning that the dither range \( r \) is dependent on the intensity value).