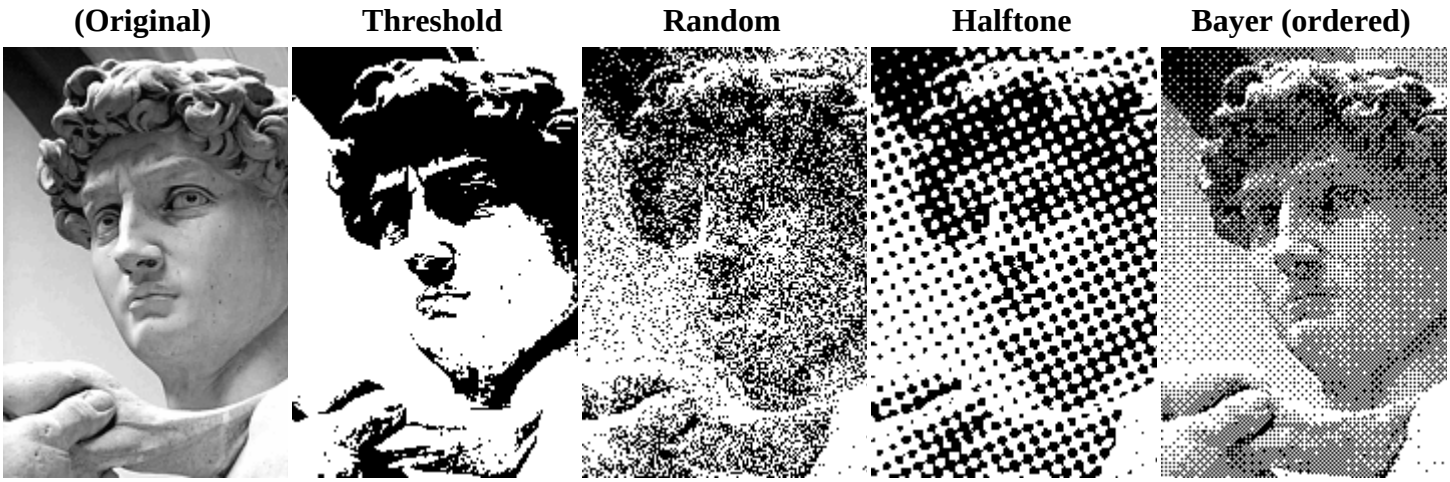
Dithering methods:

- *Thresholding* (also average dithering[14]): each pixel value is compared against a fixed threshold. This may be the simplest dithering algorithm there is, but it results in immense loss of detail and contouring.[13]
- *Random dithering* was the first attempt (at least as early as 1951) to remedy the drawbacks of thresholding. Each pixel value is compared against a random threshold, resulting in a staticky image. Although this method doesn't generate patterned artifacts, the noise tends to swamp the detail of the image. It is analogous to the practice of mezzotinting.[13]
- *Patterning* dithers using a fixed pattern. For each of the input values a fixed pattern is placed in the output image. The biggest disadvantage of this technique is that the output image is larger (by a factor of the fixed pattern size) than the input pattern.[13]
- *Ordered dithering* dithers using a "dither matrix". For every pixel in the image the value of the pattern at the corresponding location is used as a threshold. Neighboring pixels do not affect each other, making this form of dithering suitable for use in animations. Different patterns can generate completely different dithering effects. Though simple to implement, this dithering algorithm is not easily changed to work with free-form, arbitrary palettes.
    - A *Halftone dithering* matrix produces a look similar to that of halftone screening in newspapers. This is a form of clustered dithering, in that dots tend to cluster together. This can help hide the adverse effects of blurry pixels found on some older output devices. The primary use for this method is in offset printing and laser printers, in both these devices the ink or toner prefers to clump together and will not form the isolated dots generated by the other dithering methods.
    - A *Bayer matrix*[13] produces a very distinctive cross-hatch pattern.
    - A matrix tuned for *blue noise* (such as those generated by the void-and-cluster method[15]) produces a look closer to that of an error diffusion dither method.



- *Error-diffusion dithering* is a feedback process that diffuses the quantization error to neighbouring pixels.
    - Floyd–Steinberg dithering only diffuses the error to neighbouring pixels. This results in very fine-grained dithering.
    - Jarvis, Judice, and Ninke dithering diffuses the error also to pixels one step further away. The dithering is coarser, but has fewer visual artifacts. It is slower than Floyd–Steinberg dithering because it distributes errors among 12 nearby pixels instead of 4 nearby pixels for Floyd–Steinberg.
    - Stucki dithering is based on the above, but is slightly faster. Its output tends to be clean and sharp.

- Burkes dithering is a simplified form of Stucki dithering that is faster, but less clean than Stucki dithering.

Floyd–Steinberg **Jarvis, Judice & Ninke** **Stucki** **Burkes**

- Sierra dithering is based on Jarvis dithering, but it's faster while giving similar results.
- Two-row Sierra is the above method modified by Sierra to improve its speed.
- Filter Lite is an algorithm by Sierra that is much simpler and faster than Floyd–Steinberg, while still yielding similar (according to Sierra, better) results.
- Atkinson dithering, developed by Apple programmer Bill Atkinson, resembles Jarvis dithering and Sierra dithering, but it's faster. Another difference is that it doesn't diffuse the entire quantization error, but only three quarters. It tends to preserve detail well, but very light and dark areas may appear blown out.

Sierra **Two-row Sierra** **Sierra Lite** **Atkinson**