

Halftone

257522_PixelRule.tiff ↗

Halftoning is the conversion of a color or grayscale picture to a picture where only a limited number of shades are available. Most printers can only print black or white pixels, therefore halftoning is needed to convert a color or grayscale image into a 1-bit image.

Fig.15.tiff ↗

The fifteen radiobuttons at the top of the panel represent the different ways with which halftoning can be done in TIFFany II. You have the simple Threshold algorithm, a White Noise algorithm (Mezzotint), Blue Noise algorithms (Floyd-Steinberg, Jarvis-Judice-Ninke and Stucki), some Matrix-Kernel algorithms (Regular, Diag 49, Hex 41, Cluster 9, Grid 22, Triangle 16) and four Clustered Dot algorithms (Circles, Disc, Line and Square).

In the Target dpi cell you can enter the resolution of the destination picture. The destination resolution is important if you know on what kind of printer your halftoned picture will be printed. Most laser printers have 300 dpi, the NeXT laser printer has 400 dpi, professional imagesetters use 1200 or 2400 dpi.

You can choose the bit-depth of the resulting image with the Depth radio buttons. Printers usually require a 1-bit output, whereas 2-bit and 4-bit images are often used for screens with less than 24bit color depth. The Grayscale cross-button determines whether the resulting picture will be in grayscale or color.

Threshold

913680_PixelRule_Index.tiff ↵

Fig.16.tiff ↵

This algorithm just checks whether the value of each pixel is higher or lower than a given threshold. If it is higher the pixel on the generated picture will be white, otherwise it will be black. The threshold can be adjusted with the Threshold field which is visible if you activate the Threshold button.

The value you choose for threshold depends on the brightness of the picture. 50% will mostly bring satisfying results. This algorithm is especially good if you want to halftone a picture which is already nearly a black and white image (i.e. it has strong contrasts and no colors, sketches and pencil drawings etc.)

Fig.17.tiff ↵

Grayscale ramp, halftoned with Threshold, 150dpi.

Mezzotint

471795_PixelRule_Index.tiff ↵

Fig.18.tiff ↵

The Mezzotint works almost the same way as the threshold algorithm. The only difference is that the threshold value is chosen for each pixel separately using a random number. In bright areas the probability that the resulting pixel will be white is higher than in dark areas.

Fig.19.tiff ↵

Grayscale ramp, halftoned with Mezzotint, 150dpi.

Clustered Dot algorithms

9808_PixelRule_Index.tiff ↵

Fig.20.tiff ↵

The clustered dot algorithms are mostly used for medium and high resolution printing (600 dpi and higher) but may also yield good results on certain low resolution laser printers. If you choose one of the clustered dot algorithms (Circle, Disc, Line or Square) the panel will bring up a number of input boxes.

Lpi

Abbreviation for Lines per inch.
Clustered dot (pixels) are

distributed on a rectangular raster. Lpi defines the distance between the single raster lines (see illustration below).

Red/Green/Blue

Red, Green and Blue pixels all have their own raster which can be turned around its own angle. These values only have an effect on RGB images.

Alpha

If your image has an alpha channel it will be halftoned with this screen angle.

Gray

Defines around which angle the gray raster is rotated. This parameter has only an effect on Grayscale images.

Cyan/Magenta/Yellow/Black

Screen angles for CMYK images.

Fig.21.eps -

Raster angle: 30°, 4lpi

Disk and Square define the shape of the dots in the halftoned image. The pixels in a picture halftoned with Circle will be round, the ones in an image treated with Square will be square. Line will create lines of varying thickness. Circles produces concentric circles of varying

thickness.

Fig.22.tiff ↵

Grayscale ramp, halftoned with Circles, 300dpi, 10lpi (top)

Fig.23.tiff ↵

Grayscale ramp, halftoned with Disc, 300dpi, 10lpi (top)

Fig.24.tiff ↵

Grayscale ramp, halftoned with Lines, 300dpi, 10lpi (top)

Fig.25.tiff ↵

Grayscale ramp, halftoned with Square, 300dpi, 10lpi (top)

Matrix-Kernel algorithms

791315_PixelRule_Index.tiff ↵

Fig.26.tiff ↵

This kind of algorithms uses a rectangular matrix with varying threshold values. This results in a regular pattern repeating all over the image. The most frequently used of these algorithms is called Regular or Ordered Dither. This is the algorithm that NEXTSTEP

uses to display color images on a monochrome NeXT. It has the advantage of being fast and yielding good results in many cases.

Fig.27.tiff ↵

Grayscale ramp, halftoned with Regular, 150dpi

Fig.28.tiff ↵

Grayscale ramp, halftoned with Diag49, 150dpi

Fig.29.tiff ↵

Grayscale ramp, halftoned with Hex41, 150dpi

Fig.30.tiff ↵

Grayscale ramp, halftoned with Cluster9, 150dpi

Fig.31.tiff ↵

Grayscale ramp, halftoned with Grid22, 150dpi

Fig.32.tiff ↵

Grayscale ramp, halftoned with Triangle16, 150dpi

Blue Noise algorithms

764985_PixelRule_Index.tiff ↵

The way in which the algorithm decides whether a pixel in the

halftoned image will be white or black is generally the same as in the threshold algorithm. This time, though, the error (that is the aberration from the desired value) is distributed to the pixels surrounding the current pixel. If a pixel is too dark, for example the neighboring pixels will be lighted up. The three algorithms differ in the way the error is distributed to the neighborhood.

If you choose one of the Blue Noise (Floyd-Steinberg, Jarvis-Judice-Ninke, and Stucki) algorithms the panel will change to this one:

Fig.33.tiff ↵

When a picture is halftoned it may happen that undesired regular patterns appear. To prevent such effects, the first action you can take is to activate the Serpentine Raster button. Normally the lines are treated from left to right. The serpentine algorithm will take the first line from left to right, the second from right to left, the third from left to right and so on (see illustration below). This will already prevent a large part of patterns that could appear.

Fig.34.eps ↵

If this should not reduce the creation of patterns enough, you can use the Perturbation field to add noise to the resulting picture. This noise will not destroy your picture but only the unwanted patterns. Adjust the slider carefully to determine a perturbation which will preserve as much of the picture and prevent patterns. The three Blue Noise algorithms differ only slightly in their results. You will have to

try to find out which suits your pictures and needs best.

Fig.35.tiff ↵

Grayscale ramp, halftoned with Floyd-Steinberg, 150dpi

Fig.36.tiff ↵

Grayscale ramp, halftoned with Stucki, 150dpi

Fig.37.tiff ↵

Grayscale ramp, halftoned with Jarvis-Judice-Ninke, 150dpi