Floyd–Steinberg dithering

From Wikipedia, the free encyclopedia

Floyd–Steinberg dithering is an image dithering algorithm first published in 1976 by Robert W. Floyd and Louis Steinberg. It is commonly used by image manipulation software, for example when an image is converted into GIF format that is restricted to a maximum of 256 colors.

Example of a 24-bit RGB image dithered to 3-bit RGB using Floyd–Steinberg dithering

The algorithm achieves dithering using error diffusion, meaning it pushes (adds) the residual quantization error of a pixel onto its neighboring pixels, to be dealt with later. It spreads the debt out according to the distribution (shown as a map of the neighboring pixels):

$$\begin{bmatrix} \cdots & & * & \frac{7}{16} & \cdots \\ \cdots & \frac{3}{16} & \frac{5}{16} & \frac{1}{16} & \cdots \end{bmatrix}$$

The pixel indicated with a star (*) indicates the pixel currently being scanned, and the blank pixels are the previously-scanned pixels. The algorithm scans the image from left to right, top to bottom, quantizing pixel values one by one. Each time the quantization error is transferred to the neighboring pixels, while not affecting the pixels that already have been quantized. Hence, if a number of pixels have been rounded downwards, it becomes more likely that the next pixel is rounded upwards, such that on average, the quantization error is close to zero.

An image of the Statue of David, dithered with Floyd–Steinberg algorithm

The diffusion coefficients have the property that if the original pixel values are exactly halfway in between the nearest available colors, the dithered result is a checkerboard pattern. For example 50% grey data could be dithered as a black-and-white checkerboard pattern. For optimal dithering, the counting of quantization errors should be in sufficient accuracy to prevent rounding errors from affecting the result.

In some implementations, the horizontal direction of scan alternates between lines; this is called "serpentine scanning" or boustrophedon transform dithering.

In pseudocode:

```
for each y from top to bottom
   for each x from left to right
      oldpixel  := pixel[x][y]
      newpixel  := find_closest_palette_color(oldpixel)
      pixel[x][y]  := newpixel
      quant_error  := oldpixel - newpixel
      pixel[x+1][y  ] := pixel[x+1][y  ] + quant_error * 7/16
      pixel[x-1][y+1] := pixel[x-1][y+1] + quant_error * 3/16
      pixel[x  ][y+1] := pixel[x  ][y+1] + quant_error * 5/16
      pixel[x+1][y+1] := pixel[x+1][y+1] + quant_error * 1/16
```

When converting 16 bit greyscale to 8 bit, `find_closest_palette_color()` may perform just a simple rounding, for example:

```
find_closest_palette_color(oldpixel) = floor(oldpixel / 256)
```

References [edit]

- Floyd–Steinberg Dithering (Graphics course project, Visgraf lab, Brazil)

Wikipedia The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Català
Čeština
Deutsch
Español
Français
한국어
Italiano
日本語
Polski

Edit links

- R.W. Floyd, L. Steinberg, *An adaptive algorithm for spatial grey scale*. Proceedings of the Society of Information Display **17**, 75–77 (1976).

External links [edit]

- PTRANS Stand-alone ANSI-C programming language implementation.
- Matlab implementation
- iDither App
- iDitherLive App

Categories: Image processing | Computer graphics algorithms