Color Quantization

Common color resolution for high quality images is 256 levels for each Red, Green, Blue channels, or $256^3 = 16777216$ colors.

How can an image be displayed with fewer colors than it contains?

Select a subset of colors (the colormap or pallet) and map the rest of the colors to them.

**Median Cut**

![Median Cut Diagram](image1)

**Error Diffusion**

![Error Diffusion Diagram](image2)
Median Cut

Color quantization(Image, n)

For each pixel in Image with color C, map C in RGB space;

B = {RGB space};

While (n-- > 0) {
    L = Heaviest (B);
    Split L into L1 and L2;
    Remove L from B, and add L1 and L2 instead;
}

For all boxes in B do
    assign a representative (color centroid);

For each pixel in Image do
    map to one of the representatives;

The median cut algorithm

Is this algorithm image dependent?

What is the Heaviest(B) box?

Several factors have to be weighed:

- The total number of image colors in the box.
- The total number of DIFFERENT image colors in the box.
- The physical size of the box.

Which representative should be chosen for a given color?

- The representative of the box containing the color.
- The closest representative under some metric.

A better solution
Median Cut

Generalized Lloyd Algorithm - GLA

Original image 8 indexed colored

8 indexed colored 32 indexed colored
Original image

128 indexed colored

8 levels per channel
that is, 9 bits per pixel

Original Image

Threshholding

Bayer’s Ordered Dithering

Error Diffusion