COLOR-MAPPED IMAGING

Sebastiano Battiato, University of Catania, Catania, Italy
Rastislav Lukac, Epson Canada Ltd., Toronto, Ontario, Canada

Synonyms: *Color palette imaging; Pseudo-color image processing; Color quantization.*

Definition: *Imaging process devoted to work with images that store the color information of each pixel in a color table (palette).*

Brief Introduction and Terminology

A color-mapped image or indexed image manages the color information by maintaining a table with a limited number of colors [1], [2]. The information stored in each pixel is the relative index to a color in that table (Figure 1). Using indexed colors usually greatly reduces size of image files (as each pixel uses only a few bits to refer to a space in the color table, rather than many bits to refer to subtle hues). Unfortunately, it can also result in very large file sizes for photographs or images with many subtle color shades.

1	3	3	2	4
1	2	3	4	4
1	3	3	4	4
1	3	4	4	4
3	2	4	4	3

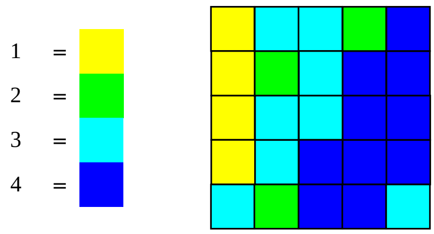


Figure 1. Example of color-mapped image.

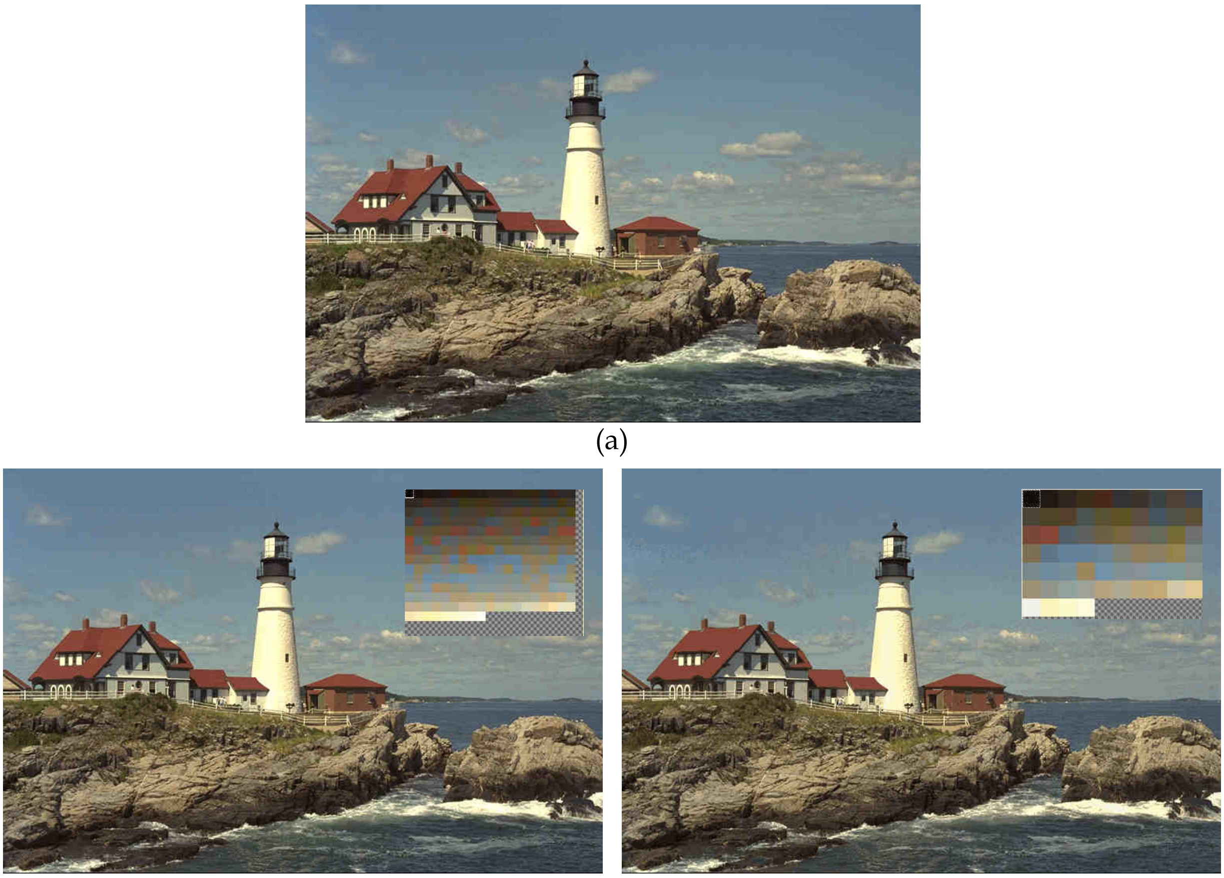
The pixels in the image I with m rows, n columns, and M distinct colors can be represented as $I(x,y) = P(I'(x,y))$, where $P = \{S_1, S_2, . . . , S_M\}$ is the set of all the colors in I, and I' is an $m \times n$ matrix of indexes in $\{1, 2, . . . ,M\}$. Typically, M is set to 16, 64 or 256. An image represented in such fashion is called indexed image (or color-mapped image) and P is its palette. The image file contains its own color palette, which is a list of the selected M colors stored in the file with the image. Each of the color in P is a red-green-blue (RGB) value properly chosen from the set of 16.7 million 24 bit colors. Thus, indexed files have 24 bits stored for each palette color, but not for each pixel. Such peculiarity allows for a substantial good visual quality of natural images even if the overall number of colors is sensibly decimated.

Color Depth and Palette Size

For too small number of available colors (i.e., small palette size), gradients and other shadings in the indexed color image can appear blocky although some ad-hoc techniques (e.g., dithering, etc.) can be used in some cases to reduce such artifacts. Image programs typically show the palette for indexed images, and allow the change of palette colors.
Given an indexed image with M colors, each pixel stores only a $\log_2 M$ bit index to specify which palette color is used. However, for almost all image formats, as specified in Table 1, it is allowed to use only 1 or 2 bytes (or integer multiple of 8 bits) for the index in I', thus limiting the real effectiveness of the representation.

Image	Image data	Palette size	Overall Size
original trucolor	768×512×24 = 9,437,184 bits	not applicable	9 Mbytes
indexed (256 colors)	768×512×8 = 3,145,728 bits	256×24 = 6144 bits	~3 Mbytes
indexed (64 colors)	768×512×5 = 1,966,080 bits	64×24 = 1536 bits	~1,87 Mbytes

Table 1. Size comparison in Mbytes between the original uncompressed version of the truecolor images shown in Figure 2a and the corresponding indexed versions shown in Figure 2b and Figure 2c.



(a) (b) (c)

Figure 2. Color-mapped imaging: (a) truecolor image, (b) indexed images having 256 colors, and (c) indexed image having 64 colors.

Figure 2 shows a 768×512 truecolor image having 29317 different colors reduced to indexed images with respectively 256 and 64 colors together with the involved palettes. Reducing the depth resolution can degrade the overall perceived quality; several artifacts could be present especially inside smooth areas (e.g., in the top area of the sky). Table 1 compares the overall size required to store these three images. The table provides a clear evidence of a suitable trade-off between the image quality and compression performances obtained by reducing the palette size.

Palette Generation

There are various ways to create the palette, to choose the proper color configuration. To reduce the number of distinct colors used in the image, usually with the intention that the new image should be as visually similar as possible to the original image, is a process commonly known as color quantization or color image quantization [3], [4]. The problem can be formalized as finding the M representative colors that will be used to reduce the number of colors, N, of the original image while reproducing the original image with the highest possible fidelity. Color reduction algorithms can be used in various kind of display hardware, including LCDs used in mobile phones, capable of showing a smaller color range than more advanced displays [5]. The visibility of artifacts due to the quantization errors can be sensibly reduced by making use of proper algorithms and techniques devoted to pick-up the most suitable set of colors. With the few colors available on early computers, different quantization algorithms produced very different-looking output images. Fortunately, recent advances in imaging hardware and software together with refined color-mapping algorithms have allowed for an output indistinguishable from the view through the camera lens given a sufficiently large palette.

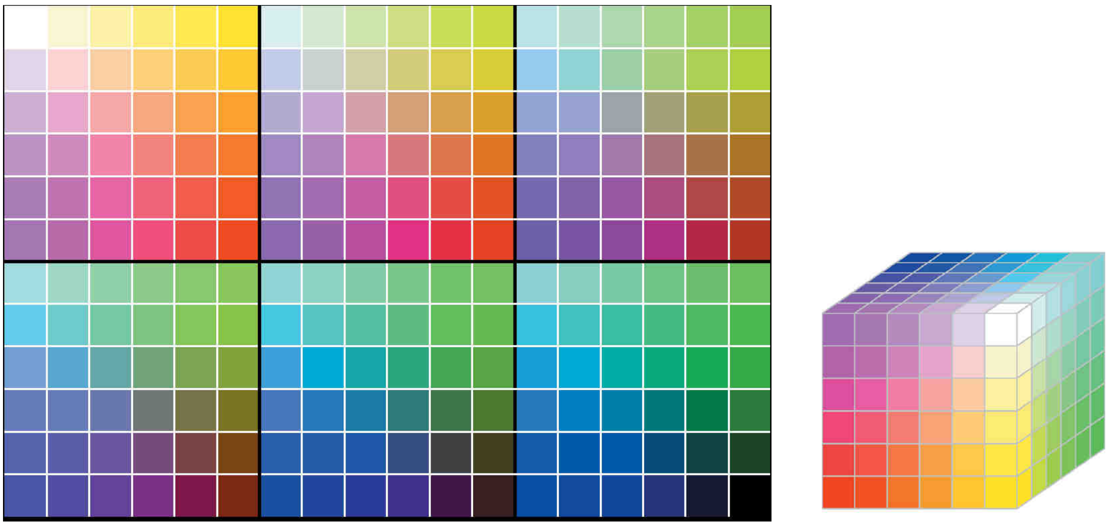


Figure 3 . *All-systems-safe* colors.

If the palette is a priori fixed, color quantization is usually done using the so-called straight-line distance or nearest color algorithm. The algorithm takes each color in the original image and finds the closest palette entry, minimizing the distance between the two corresponding points in three-dimensional color space. Although a number of needed colors may not be available in the fixed-color palette, and many of the available colors may not be used; there are, however, numerous applications in which it simply makes no sense to use more than a few hundred, and sometimes fewer, colors. Given the variety of current imaging systems, it is of considerable interest to have a subset of colors that are likely to be reproduced faithfully, reasonably independently of viewer hardware capabilities. This subset of colors is called the set of *all-systems-safe* colors. In Internet applications, they are called safe Web colors or safe browser colors. On the assumption that 256 colors is the minimum number of colors that can be reproduced faithfully by any system in which a desired result is likely to be displayed, it is useful to have a unified framework. Forty of these 256 colors are known to be processed differently by various operating systems, leaving only 216 colors that are common to most systems. These 216 colors have become a de facto standard for safe colors, especially in Internet applications just to guarantee that the colors viewed by most people appear the same. Each of the 216 safe colors is formed using RGB triplets, however, each of the triplets' components can only be 0, 51, 102, 153, 204, or 255, thus providing 6^3=216 possible color

combinations (note that all values are divisible by 3). It is customary to express these values in the hexagonal number system. Figure 3 shows 216 safe colors arranged in a descending order. The square in the top left array has value FFFFFF (white). The final (bottom right) square of the last array has value 000000 (black). It is important to note that not all possible 8-bit gray colors are included in the 216 safe colors. Figure 3 also shows the RGB safe-color cube. Unlike the full-color cube which contains different color combinations inside and on its surface, the RGB safe-color cube has valid colors only on the surface planes. Each plane consists of 36 colors, so the entire surface of the safe-color cube is covered by 216 different colors, as expected.
An optimized color palette is one in which the available colors are chosen based on the frequency of their occurrence in the original source image. Popular modern algorithms for optimized palette generation include the median cut algorithm a technique based on octrees [6]. More advanced techniques can be found in [4],[7],[8],[9].
Once the set of colors has been derived it is possible to apply some techniques to achieve greater compression ratio just reducing the local index redundancy [10],[11],[12]. The so-called reindexing techniques aim at finding the optimal reordering without the consideration of all possible color indexing combinations (M! combinations for an image with M colors).

Hardware/Software Palettes

In some cases, only a predefined palette is available due to the hardware/software constraints and capabilities (dictated by the hardware/software palette). In this case the limited number of colors is also related to the number of available bits used for the corresponding indexes. The list of the full-color palettes (with some examples) for notable computer graphics hardware, together with a detailed description of the software palettes used in many personal computers of early 1990s (e.g., Apple, Commodore, Sinclair, IBM/PC compatible systems, Atari, etc.) are presented in [5].
Another important usage of a priori known palettes is in the field of data visualization. In that case input image is processed just to map the original input pixel value to a given color. Several predefined palettes are known and available, although their effectiveness is strictly related with the underlying imaging field (e.g., medical, remote sensing, industrial inspection, etc.). Popular imaging tools such as ImageJ (http://rsb.info.nih.gov/ij/) or MATLAB (http://www.mathworks.com/) offer a great variety of color palettes. Figure 4 shows a gray level image, represented with two different color palettes able to highlight different details.
The corresponding color palette can also be derived by using some automatic techniques, such as intensity slicing and intensity to color transformation. See [1] for details.

(a) (b) (c)

Figure 4. Pseudo color imaging: (a) original gray level image, and (b, c) indexed images with 256 colors obtained by using two different color palettes devoted to highlight different areas of the input images. The palette used in (b) is a "hot" palette, whereas in (c) is a "cool" palette.

Managing Color Quantization Error

It is common to combine color quantization with some heuristics to create an impression of a larger number of colors and eliminate banding artifacts. Dithering is a technique used in computer graphics to create the illusion of color depth in images with a limited color palette. In a dithered image, colors not available in the palette are approximated by a diffusion of colored pixels from within the available palette. The human eye perceives the diffusion as a mixture of the colors within it. Dithering is analogous to the halftone technique used in printing [13]. Dithered images, particularly those with relatively few colors, can often be distinguished by a characteristic graininess, or speckled appearance [14]. Figure 5 shows an example of Floyd-Stenberg dithering [15] applied to a truecolor image compared with the corresponding 16 color image without dithering.

(a) (b) (c)

Figure 5. Dithering in color-mapped imaging: (a) truecolor image with 26124 colors, (b) indexed image with 16 colors obtained without using advanced color reduction techniques, (c) masking the band artifacts using a dithering technique in [15].

Image Formats

Color-mapped imaging is mainly used in Graphics Interchange Format (GIF) and Portable Network Graphics (PNG) images [16].

GIF, for a long time the most popular lossless and animated bitmap format on the World Wide Web, only supports up to 256 colors, necessitating quantization for many images. GIF images are compressed using the Lempel-Ziv-Welch (LZW) lossless data compression technique to reduce the file size without degrading the visual quality. GIF images can have transparency. This is useful for maintaining the color background (e.g., web page background). GIF images can be interlaced meaning that a low-quality version of the picture appears first, and then it gets progressively drawn over by the final image.

PNG was created in 1995 as a response to Unisys' enforcement of their patent on the LZW compression algorithm used in GIFs. Although Unisys' patent has expired, a number reasons for switching to PNGs from GIFs remain. PNG supports palette-based (palettes of 24-bit RGB colors), grayscale, and RGB color images. Since PNG files use fewer bits per pixel for palettized images, PNG images can often be made much smaller in file size without much visual degradation by application of color quantization than GIF images. An image in a lossless PNG file can be 5%-25% more compressed than a GIF file of the same image. Saving, restoring and resaving a PNG image will not degrade its quality. PNG does not support animation, as opposed to GIF. Thus, the PNG format is more suitable than GIF, especially in instances where truecolor imaging, alpha transparency, or lossless data formats are required.

See: Reindexing techniques, GIF format, Median cut algorithm, Dithering.

References

1. R.C. Gonzalez, R.E. Woods, "Digital Image Processing", Third Edition, Prentice Hall, 2007.
2. Foley, vanDam, Feiner, Hughes, "Computer Graphics", second edition, Addison Wesley, 1990.
3. M.T. Orchard, C.A. Bouman, "Color Quantization of Images", IEEE Transactions on Signal Processing, vol. 39, no. 12, pp. 2677-2690, December 1991.
4. J. Puzicha, M. Held, J. Ketterer, J.M. Buhmann, D.W. Fellner, "On Spatial Quantization of Color Images", IEEE Transactions on Image Processing, vol. 9, no. 4, pp. 666-682, April 2000.
5. List of palettes – Wikipedia (http://en.wikipedia.org/wiki/List_of_palettes) , 2007.
6. Heckbert P.: Color Image Quantization for Frame Buffer Display. ACM Comp. Graphics, SIGGRAPH, vol. 16, no. 3, pp. 297–307, July 1982.
7. N. Papamarkos, A.E. Atsalakis, C.P. Strouthopoulos, "Adaptive Color Reduction", IEEE Transactions on Systems, Man and Cybernetics, Part B, vol. 32, no. 1, pp. 44-56, February 2002.
8. I. Ashdown, "Octree color quantization," Radiosity—A Programmer's Perspective, Wiley, 1994.
9. S.C. Pei, Y.T. Chuang, W.H. Chuang, "Effective Palette Indexing for Image Compression Using Self-Organization of Kohonen Feature Map", IEEE Transaction on Image Processing, vol. 15, no. 9, pp. 2493–2498, September 2006.
10. A. Pinho and A. Neves, "A Survey on Palette Reordering Methods for Improving the Compression of Color-indexed Images", IEEE Transactions on Image Processing, vol. 13, no. 11, pp. 1411-1418, November 2004.
11. S. Battiato, G. Gallo, G. Impoco, F. Stanco, "An Efficient Re-indexing Algorithm for Colour-Mapped Images", IEEE Transactions on Image Processing, vol. 13. no. 11, pp. 1419-1423, November 2004.
12. S. Battiato, F. Rundo, F. Stanco, "Self Organizing Motor Maps for Colour Mapped Image Re-indexing, IEEE Transactions on Image Processing", vol. 16, no. 12, pp. 2905-2915, December 2007.
13. R. Ulichney, "Digital Halftoning", Cambrdige, MA, MIT Press 1987.
14. J. M. Buhmann, "Dithered Color Quantization," in Proc. EUROGRAPHICS Lisboa, Portugal, Computer Graphics Forum, vol. 17, no. 3, pp. 219-231, 1998.
15. Floyd R. W., Steinberg L., "An Adaptive Algorithm for Spatial Gray Scale", In Proceedings of ID (Society for Information Displays) – Digest of technical papers , pp. 36-37, 1975.
16. J. Miano, "Compressed Image File Formats", ACM Press, Addison-Wesley Professional, 1999.
17. G. Roleof, "PNG: The Definitive Guide", Second Edition, http://www.libpng.org/pub/png/book/, 2003.