Topical Meeting on Applied Vision, 1989 Technical Digest Series, Vol. 16,
(Optical Society of America, Washington, D.C., 1989)

Challenges in Device-Independent Image Rendering

Robert A. Ulichney

Digital Equipment Corporation
550 King St., LKG1-2/C13, Littleton, MA 01460-1289

Abstract

At display or print time, images must be properly scaled with optional sharpening, tone-scale or color adjusted, and quantized either dynamically or simply, depending on the available color levels at the targeted device.

1. Introduction

The goal of incorporating images as a generic data type (like ASCII text) in general purpose computer systems poses a considerable design challenge. The user has access to wide range of display devices that vary dramatically in resolution and size and color or gray-scale quantization values. It should be a transparent operation to send an image to an arbitrarily shaped work-station window, a terminal with asymmetric pixels, or a bitonal hard copy device, and have the fidelity of the original preserved as best as it can be.

This paper addresses this problem from a systems point of view and presents design tradeoffs that exist. Perhaps the most difficult challenge is that of gamut mapping between devices; to complement the other papers in this session that focus on this topic, this paper will instead concentrate on the other parameters that must be satisfied for device-independent rendering to work.

It should be clarified that in this paper the use of the term "rendering" should not be confused with the sense used in 3-D graphics synthesis.

2. Rendering Model

An overview of a system for image rendering is shown is the figure. The task of the "Render" block is take decompressed image data and tailor it for a specific target display. Successful rendering requires intimate knowledge of the nature of the display device; this can be modeled as the Physical Reconstruction Function [1, ch. 2].

With the soon-to-be-adopted DCT-based international standard for continuous-tone color image compression [2], it is reasonable to assume that source images will be stored so compressed in a "pristine" device-independent way. The acceptance of a standard will hopefully mean readily available hardware to quickly perform decompression.

The rendering system comprises the three stages of (1) Resample/Sharpen, (2) Color/Tone Adjust and (3) Dynamic or Simple Quantization. In the first stage, the original must be resam-pled to match the grid of the target. Most often this will mean a simple digital enlargement or reduction; however, the possibility exist for a change in pixel aspect ratio, or even a rectangular to hexagonal lattice conversion. For simple rectangular-to-rectangular grid scaling, the best filters to use have been determined from a perceptual point of view [3]. When bandlimiting

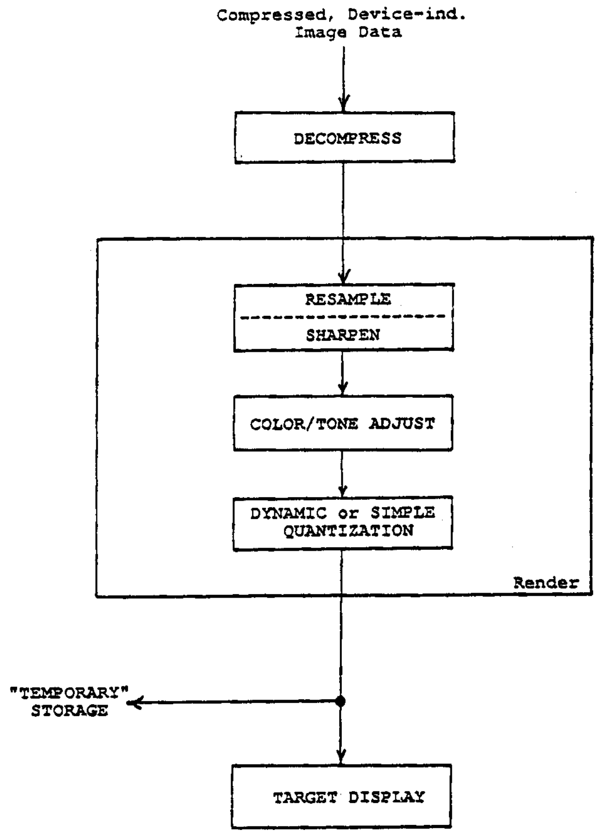

Image rendering system. Parameters are tailored for each target display.

for reduction, a Gaussian with $\sigma = 0.30 \times \text{output\_period}$ is recommended. For interpolation, it was found that the filter most preferred in terms of looking the most like the original was a cascade of two: first sharpen with a Laplacian followed by convolution with a Gaussian with $\sigma = 0.375 \times \text{input\_period}$.

A typical sharpening scheme can be expressed as follows:

$$J_{\text{sharp}}[x, y] = J[x, y] - \beta \Psi[x, y] * J[x, y].$$

where $J[x, y]$ is the continuous-tone input image, $\Psi[x, y]$ is a digital Laplacian, and "$*$" is the convolution operator. The nonnegative parameter β controls the degree of sharpness. Besides the need for sharpening in interpolation, it belongs in the rendering model for other reasons. If sharpening is desired on an image that is being reduced in scale, it must occur *after* scaling and thus cannot be done outside of the rendering system. Also, if dithering is later used as the quantization means, it may be needed to compensate for a loss of high spatial frequencies.

Convolutions are expensive. For scaling up or down by factors less than 2.5, nearest neighbor (convolutionless) scaling is usually adequate.

The problem of gamut mapping is handled in the second stage. Again, this difficult topic is addressed by the other papers in this session. For the case of monochromatic displays, a one-dimensional mapping can be established by direct measurement of a test gray ramp.

When sufficient color quantization levels exist so as to avoid spurious contours, a simple uniform quantizer is all that is needed in the third stage of the rendering system. Perhaps most commonly, however, is the case where the targeted device has an insufficient number of colors or frame-buffer memory requiring what can be called "dynamic quantization".

3. Dynamic Quantization

There are three basic classes of techniques for circumventing the problem of insufficient colors or color memory: (1) histogram-based methods, (2) chrominance subsampling, and (3) dithering.

Histogram-based methods all require two passes of the entire image data; the first to acquire the histogram statistics from which a 3-D quantizer to N colors is fabricated, and a second to perform the pixel assignments. Perhaps the fastest method is the popularity algorithm [4] where a simple sort finds the N colors with the highest frequency, and all other colors are mapped to those.

A more compute intensive method but one that in general performs much better is the often used median-cut algorithm [4]. In this method the color space is repeatedly subdivided into smaller rectangular solids at the median planes, the goal being that each of the selected colors represent an equal number of colors in the image. The average of the colors in each of the final regions are the colors used in the quantizer. A later, less compute-intensive variation on this is the mean-split algorithm.

More recently, a clustering technique has been reported [5] that has less quantization error than the above mentioned methods, as minimization of sum-of-squared-errors is central to its design.

Chrominance subsampling [6,7,8] exploits the perceptual fact that chrominance acuity is much less than that for luminance. Typical implementations average each of the two chrominance values in a given luminance-chrominance color representation over either a 2×2 or 4×4 region; this results in an average of 12 or 9 bits per pixel respectively assuming 8 bits of amplitude resolution per component. This approach requires hardware, usually at video rates, to up-sample the chrominance components and convert the color space. The tradeoff is special purpose hardware for less frame-buffer memory.

The third alternative for dynamic quantization is a dithering method. Several methods exist [9,10,11,1], primarily designed for binary output but all extendible to multilevel color. The basic principle is to use the available subset of colors to produce the illusion of any color in between by judicious arrangement.

The computationally simplest method is the point process of ordered dither where a deterministic "noise" array tiles the plane in a periodic manner. Each component of each pixel in the image has associated with it a "noise" or dither amplitude that is added to it before being passed to a uniform quantizer. Images on most electronic devices look best with what is called a "dispersed-dot" dither pattern, patterns with take best advantage of the available spatial resolution to produce homogeneous distributions of noise. Some bitonal devices cannot accommodate isolated dots so a clustered-dot such as the printer's screen works best in that case.

Because the patterns are periodic, ordered dither does suffer from the low frequency texture associated with the periodicity. Where additional computation can be afforded the aperiodic operation of error diffusion-based techniques produce pleasing results. Error diffusion uses a uniform quantizer but distributes the quantization error for each pixel over some neighborhood of yet-to-be-quantized pixels.

Combinations of these quantization alternatives can be used but the increase in computation begins to become less fruitful in terms of quality increase.

4. Architectural Concerns

The default parameters for such a rendering system should be chosen so as to best preserve the integrity of the source image. When the user needs to change the image from these defaults, controls should be supplied for, say, tint- and luminance-control, sharpness, and scale changes. But what processing time per change should the system be designed for? (Speed is not that critical for hard-copy rendering, but is in the workstation environment.) Is .5 seconds necessary or can 10 seconds be tolerated? Such requirements have enormous cost consequences as the choice is between expensive special purpose hardware or a general purpose processor-based software solution.

If rendering speeds are inhibitive for certain applications where the same images are repeatedly displayed on the same device, it is important to allow the device-dependent rendered image to be saved in what is called "temporary" storage in the figure.

Making an image data type easy to use depends on a workable device-independent rendering strategy; this is perhaps the next fundamental challenge in computer system design.

References

[1] Ulichney, R.A., *Digital Halftoning*, Cambridge: The MIT Press (1987).

[2] Hudson, G.P., H. Yasuda and I. Sebestyen, "The international standardization of a still picture compression technique," IEEE Global Telecomm. Conf., pp. 1016–1021, Nov 28–Dec 1 (1988).

[3] Schreiber, W.F. and D.E. Troxel, "Transformation between continuous and discrete representation of images: a perceptual approach," *IEEE Trans. PAMI*, vol. PAMI-7, no. 2, pp. 178–186, (1985).

[4] Heckbert, P.S., "Color image quantization for frame buffer display," *Computer Graphics* (AMC SIGGRPAPH'82 Conf. Proc.), vol. 16, no. 3, pp. 297–307 (1982).

[5] Wan, S.J., K.M. Wong, and P. Prusinkiewicz, "An algorithm for multidimensional data clustering," *ACM Trans. on Math. Software*, vol. 14, no. 2, pp. 153–162 (1988).

[6] Sigel, C., R. Abruzzi, J. Munson, "Chromatic Subsampling for display of color images," *—this conference—* (1989).

[7] Luther, Arch, *Digital Video in the PC environment*, New York: McGraw-Hill, pp. 193–194 (1989).

[8] Glass, L.B., "Digital Video Interactive," *Byte*, May, p. 284 (1989).

[9] Roetling, P.G., "Binary approximation of continuous-tone images," *Photographic Science and Engineering*, vol. 21, pp. 60–65 (1977).

[10] Stoffel, J.C. and J.F. Moreland, "A survey of electronic techniques for pictorial reproduction," *IEEE Tran. Commun.*, vol. 29, 1898–1925 (1981).

[11] Jarvis, J.F., C.N. Judice, and W.H. Ninke, "A survey of techniques for the display of continuous-tone pictures on bilevel displays," *Computer Graphics and Image Processing*, vol. 5, pp. 13–40 (1976).