A Review of Halftoning Techniques

Robert Ulichney

Cambridge Research Lab, Compaq Computer Corp., 1 Kendall Sq., Cambridge, MA 02139, USA[†]

ABSTRACT

Digital halftoning remains an active area of research with a plethora of new and enhanced methods. While several fine overviews exist, this purpose of this paper is to review retrospectively the basic classes of techniques. Halftoning algorithms are presented by the nature of the appearance of resulting patterns, including white noise, recursive tessellation, the classical screen, and blue noise. The metric of radially averaged power spectra is reviewed, and special attention is paid to frequency domain characteristics. The paper concludes with a look at the components that comprise a complete image rendering system. In particular when the number of output levels is not restricted to be a power of 2. A very efficient means of multi-level dithering is presented based on scaling order-dither arrays. The case of real-time video rendering is considered where the YUV-to-RGB conversion is incorporated in the dithering system. Example illustrations are included for each of the techniques described.

Keywords: Dithering; thresholding; ordered dithering; error diffusion; white noise; blue noise; rendering systems.

1. INTRODUCTION

Halftoning is the process of transforming an image with greater amplitude resolution to one with lesser amplitude resolution. This has been practiced for over a hundred years in the printing industry: the solution for displaying continuous-tone images with only black or white dots. In digital printing and display systems today the problem is essentially the same: how to produce the illusion of the original tonal quality of a image by judicious placement of dots. The goal of all halftoning techniques is to generate an image with fewer amplitude levels that is perceptually similar to the original.

The "perceptually similar" criterion is a very important one, and is linked to the characteristics of human vision. If we take a more traditional engineering criterion of minimizing mean-square-error, for example, a halftoning system would generate the output shown in Figure 1 of a continuous tone grayscale photograph. Since human vision is the mechanism for measuring halftone output performance other approaches must be taken. We exploit the fact that our high spatial frequency response drops quickly above 6 or so cycles per degree. So high frequency patterns are perceived as their macroscopic average.

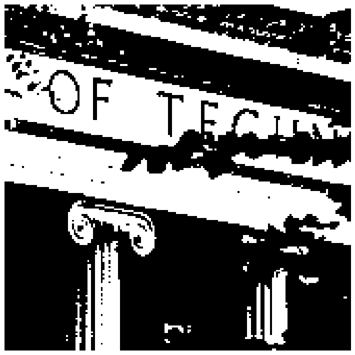
Figure 1. Minimum MSE output: result of a fixed threshold.

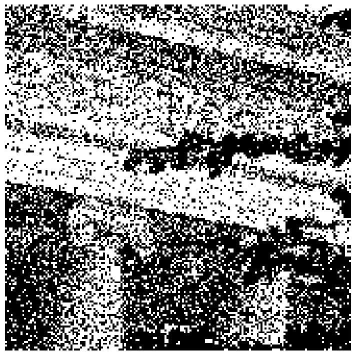
Figure 2. The result of dithering with a white noise threshold.

[†] For further author information – http://www.crl.research.digital.com/who/people/ulichney/bio.htm; Email: ulichney@crl.dec.com; Tel: 617-551-7633.

The field of halftoning is an active area of research with several fine reviews recently available. Last year at this conference, Keith Knox presented an excellent overview [1]. Reiner Eschbach compiled a collection of papers [2] covering a cross-section of activity in this area, and a new book on the field was just released by Henry Kang [3]. The purpose of this paper is to simply serve as a review by identifying the basic classes of halftone patterns based on appearance.

Dithering systems can be used to process color images on display devices with the capability to handle more than 2 levels of amplitude, as will be address at the end of this article. It is easier, however, to demonstrate and conceptualize dithering methods when used for generating bitonal output, and so examples will be shown for this case. We will explore various key classes of techniques for achieving the halftoning "illusion" and their tradeoffs.

2. WHITE NOISE

Perhaps the first solution that comes to mind when considering the problem of how to distribute pixels to form macroscopic averages is to threshold with white noise. Historically, this was the first approach taken [4] for electronic displays with independently addressable dots. An example of this is shown in Figure 2. While the technique will break up the spurious contours seen in Figure 1, it suffers from apparent graininess. This is due to the presence of long wavelengths in the dither patterns.

2.1 Frequency-Domain Metric

Representing signals in the frequency domain can often simplify complexity seen in the spatial domain. This is indeed the case with dither patterns. It allows us a means to examine the distribution of energy and its consequences on the quality of the patterns. As it is the flat regions of an image where the nature of dither is most important, the focus will be on the power spectrum of patterns that result from the dithering of a single fixed gray level.

However two-dimensional spectral plots can also afford to be made more succinct. Most well formed patterns share the characteristic of being isotropic. This leads to a metric that summarizes the spectrum radially. Figure 3 shows the segmenting of a spectral period into concentric annuli. Averaging the power spectrum in each annulus results in a "radially averaged power spectrum" where these averages are plotted against the "radial frequency", the radial distance from the DC center point to the annulus. As with the horizontal or vertical spatial frequency, this radial frequency is in units of inverse spatial sample periods.

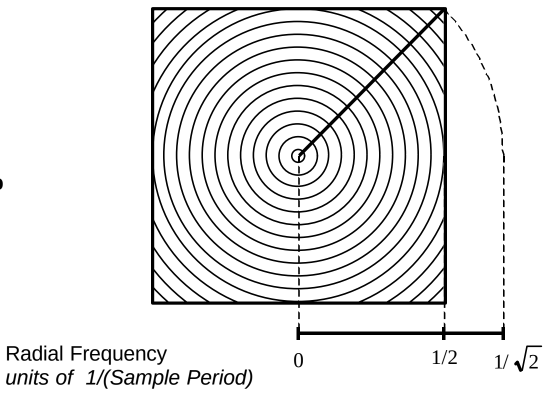

Figure 3. Segmenting the Fourier transform into concentric annuli.

The DC or zero frequency center point corresponds to the macroscopic average, or gray level, of the dither pattern. As this datum is already known, it contributes nothing to aiding our interpretation of the nature of the distribution of pixels that make up the dither pattern, and so is omitted in resulting plots. Radial frequency can go as high as $1/\sqrt{2}$ at the high frequency corners; these high frequency corners correspond to a checkerboard pattern, the highest 2-D pattern possible.

The power spectrum also needs to be normalized for the gray level, g, which we will specify as ranging from g=0 (black) to g=1.0 (white); normalizing in this way separates gray level from the specific number of bits used in the input image. Spectral energy increases as the number of minority pixels in a bitonal pattern increases, peaking at g=0.5, and so the spectral values are divided by the gray level variance [6, ch. 3]

$$s_g^2 = g(1-g)\,.$$

Figure 4 shows the measured radially averaged power spectrum for white noise patterns. Using the normalization described above, the same plot results for all gray levels, differing only in the small perturbations around 1.0. As expected, dithering with a white noise threshold produces patterns with a flat spectrum. The term “white noise” falls from the fact that equal energy is present at all wavelengths. This includes energy at low frequencies, which is the cause of the grainy artifacts in Figure 2. While the higher frequencies tend to be invisible, the arbitrarily long wavelength of the low frequencies can be very noticeable and detract from the image content.

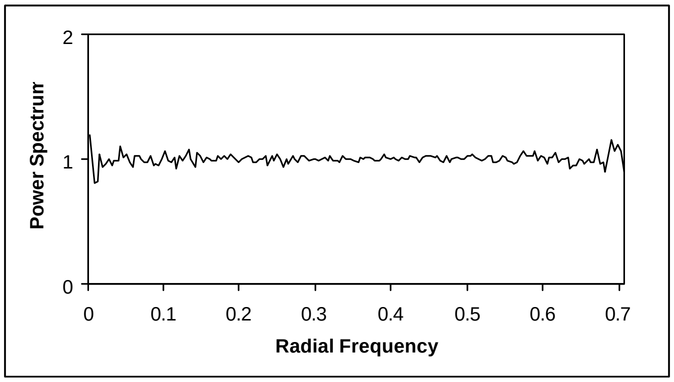

Figure 4. Radially averaged power spectrum for white noise dither patterns for all gray levels.

3. CLUSTERED-DOT

Clustered-dot halftones are those that we commonly see in mass hard copy publications produce by offset printing, such as in newspapers, magazines and books. As opposed to dispersed-dot patterns, where the addressability of each individual pixel is used, the pixels in clustered-dot patterns are nucleated in groups in regular intervals. Figure 5 illustrates an example of this type of halftone. It is important to note that the pixel size of this (and other) examples is much larger than practical printing systems, and is shown as such to allow detailed examination.

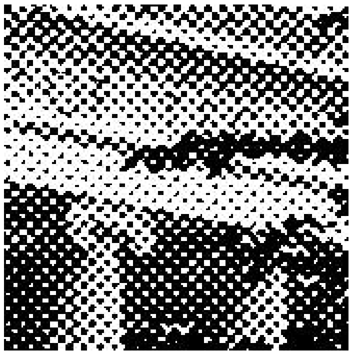

Figure 5. Result of halftoning with a 4x4 super-cell classical screen.

Around 1850, the feasibility of a process for printing continuous-tone images was demonstrated by photographing an image through a loosely woven fabric or “screen” placed some distance from the focal plane. It is this process that gave us the word “halftone”. It came into practical use in the 1890's when the halftone screen became commercially available, consisting of two ruled glass plates cemented together. In the 1940's the contact screen, a film bearing a properly exposed light distribution of a conventional screen, was introduced.

This screen is sometimes called the graphic arts screen, printer's screen, or classical screen. Even in view of the popularity of dispersed-dot screens, this type of screen is still very important for many forms of printing. In the case of offset printing, there is a minimum size dot that will hold ink, so high-resolution dispersed-dot screens will not work. In electrophotographic (laser) printing a clustered-dot screen is more robust against fluctuations in density. Since the visual system is most acute for orientations at 0 and 90 degrees and least acute at 45 degrees [5], using a screen oriented at 45 degrees was empirically determined as optimal from the earliest days of halftoning.

To build a dither template that will produce a classical screen digitally, the ordering suggested by Figure 6 is used. The dither template is a matrix of integers specifying the order in which pixels will be turned “on” as gray level increases. It is periodic and is thus replicated throughout all of two-space. In this figure, the array is segmented into four regions. Threshold value ordering begins with at the centers of the dark spirals until half of the array elements are assigned, then continues from the outsides of the dotted spirals until the remain half is assigned.

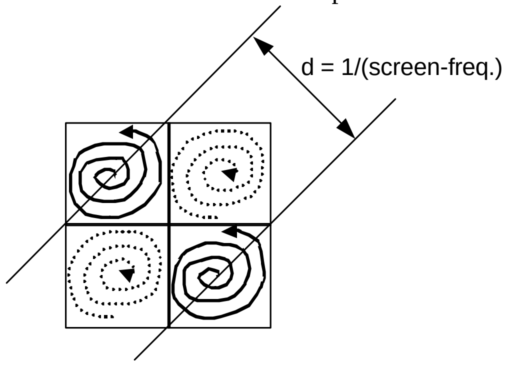

Figure 6. Ordering strategy for digitally generating a classical screen.

13	11	12	15	18	20	19	16
4	3	2	9	27	28	29	22
5	0	1	10	26	31	30	21
8	6	7	14	23	25	24	17
18	20	19	16	13	11	12	15
27	28	29	22	4	3	2	9
26	31	30	21	5	0	1	10
23	25	24	17	8	6	7	14

Figure 7. Dither templates for a 4x4 super-cell classical screen.

A characteristic of the classical screen is the screen frequency, usually stated in terms of 45-degree screen-lines per inch (lpi). Newspapers typically use 85 lpi screens, while glossy-paper magazines will use 150 lpi screens. Figure 6 indicates how the screen frequency is related to the dither template. The screen period *d* depends on the number of pixels in the macro-cells and on the resolution of the final output. By way of example, consider a classical screen with 8x8 macro-cells printed on a 1200 dots-per-inch printer. In this case $d = 8\sqrt{2}/1200$, and the screen frequency would be 1/*d* or about 106 lines-per-inch.

The dither template for an 4x4 macro cell classical screen is shown in Figure 7. This is called a “template” because the actual dither arrays must be normalized to the number of input levels, as detailed in the last section of this article. Figure 5 was generated with the dither template shown in Figure 7.

If we take an 8x8 macro cell and use it to dither several fixed gray levels, we can use the radially averaged power spectrum to examine its frequency domain characteristics. A plot of spectra for this case is shown in Figure 8. As stated earlier, the DC term is omitted because it simply reflects the average gray level and does not contribute to nature of the dither pattern under observation. In this case we see a preponderance of low frequency energy for all gray levels. This is consistent with what we would expect for a clustered dot.

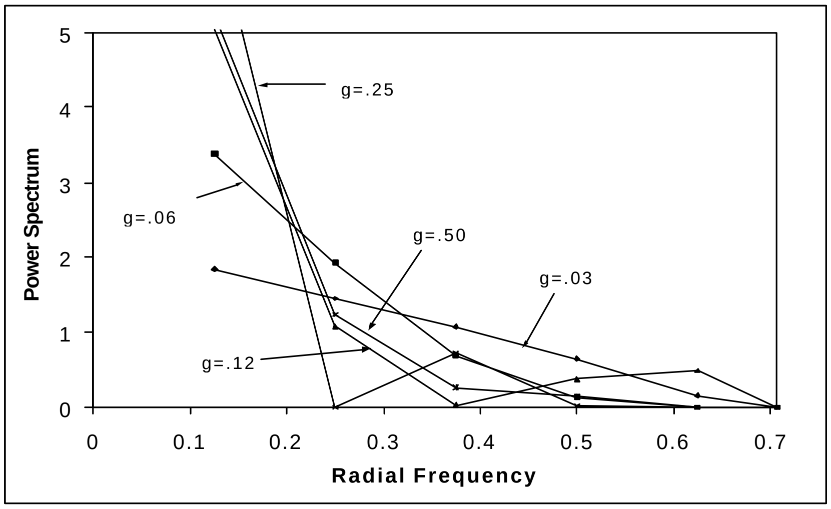

Figure 8. Radial spectra of gray levels dithered with an 8x8 super-cell classical screen.

Recall that in the case of white noise, it was the low frequencies that contributed to graininess. So, in this case we might assume that clustered-dot patterns will always suffer from low-frequency textures. It is important to point out that the unit of radial frequency in these plots is inverse sample period. Not counting the DC component, the lowest frequency component of a periodic dither pattern is a function of the dither template size and the spatial sample period (resolution) of the display device. If the dither template size is small enough and the spatial sampler period is high enough, then this lowest frequency will still be a fairly high frequency in absolute terms. It is under these conditions that clustered-dot halftoning work well.

In offset color printing, images are produced by overlaying component images in each of four printing inks, CMYK: Cyan, Magenta, Yellow and blacK. To avoid moiré patterns, the clustered-dot screen for each of the component images are designed at different angles. The black screen, the most apparent color, is set at 45°, the least apparent color, yellow, is set at 0°, and cyan and magenta are set at $\pm 15^\circ$.

4. RECURSIVE TESSELLATION

For displays or printers that are not of the highest spatial resolution, it is desirable to use dispersed-dot dither patterns. A historically popular choice for dispersed-dot dither templates have been those that can be generated by the method of recursive tessellation. The goal is to create a template ordering so dither patterns that result from periodically replicating the fundamental period is as homogeneous as possible.

Figure 9 shows the stages in recursively tessellating or tiling the plane to construct the ordering of a 4x4 dither template. In stage i=1, shown in part (a), the fundamental 4x4 period is identified and replicated throughout all of two-space. Note that the top and bottom edges of the gray period shown are copies of each other, as are the left and right edges. We begin at the center position and assign a rank of 0. This selection is periodically replicated. With the goal of homogeneity, the candidate for the next position should be in the center of the voids between the replicated positions already assigned. This void center can be found by constructing perpendicular bisectors between nearest neighbors; the corners of the resulting tile are the next candidates, and they are thus labeled with the next rank, 1.

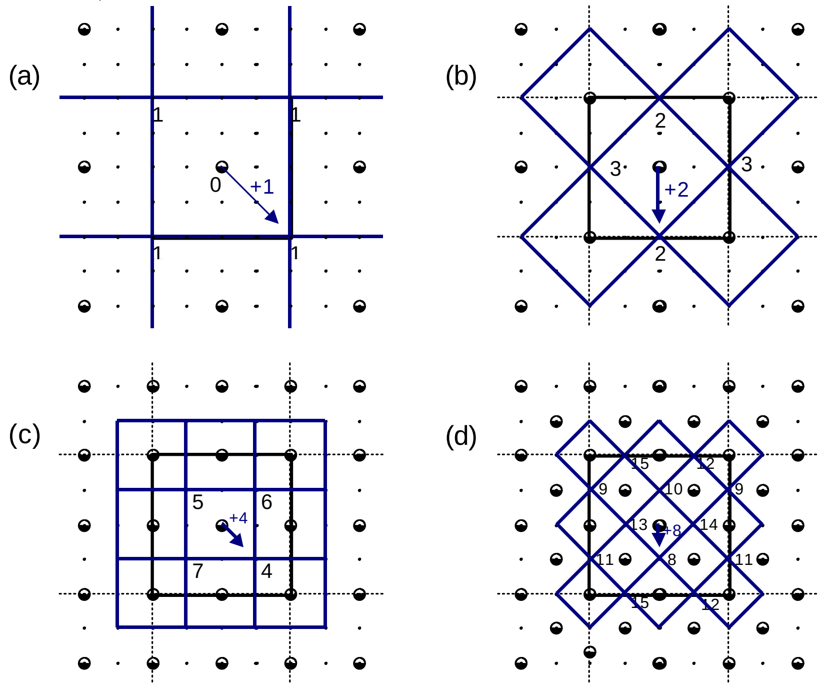

Figure 9. Stages in generating a recursive-tessellation dither template.

In stage i=2, in part (b), void centers are again found at the corners of sub-tiles formed by perpendicularly bisecting nearest neighbors of already assigned points. The ranks of these new corner points are assigned by summing the rank of the point in the center of the sub-tile and 2^{i-1}, where i is the current stage. This summing is depicted in the vector as shown. The direction of the vector from the sub-tile center to a corner can be to any of the four possibilities, but must remain fixed from tile to tile.

The process continues for stages i=3, in part (c), and i=4, in part (d) completing the assignment of all 16 elements. The completed 4x4 dither template is shown is Figure 10, and an image dithered with this array is shown in Figure 11. A recursive tessellation array of this size is said to be 4th order [6, ch. 6]. An array of order η will have 2^η unique elements.

Dithering with these type of dither template is referred to as Bayer's dither, in reference to his famous 1973 proof of optimality [7]. Also, dithering with these arrays, as well as those used for clustered-dot halftoning, are part of a larger genre referred to as ordered dither. Ordered dither is the name given to any dither process that uses a period deterministic dither array. It is the “ordered” nature of the elements in the array that contrast it with the random nature of white noise dithering.

0	14	3	13
8	4	11	7
2	12	1	15
10	6	9	5

Figure 10. Values of a 4th-order recursive tessellation template.

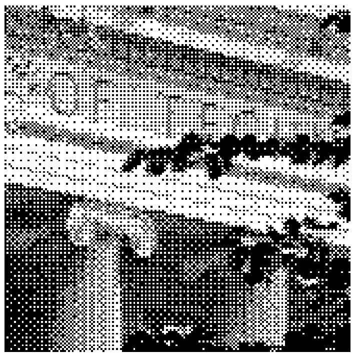

Figure 11. Dithering with a 4th order recursive tessellation array.

5. BLUE NOISE

Engineers like to describe various types of noise with color names. The most well known example is “white noise”, so named because its power spectrum is flat across all frequencies, much like the visible frequencies of light. “Pink noise” is used to describe low frequency white noise, the power spectrum of which is flat out to some finite high frequency limit. The spectrum associated with Brownian motion is (perhaps whimsically) referred to as “brown noise”.

Blue noise is the high frequency compliment of pink noise. As we have seen, when it comes to producing good quality halftone patterns, low frequency energy is the enemy. In this section the concept of the blue noise metric is described, along with neighborhood and point processes that generate blue noise dither patterns. Since its introduction [8], the blue noise concept has become an important part of halftoning research.

Ideally, a well-formed dither pattern should have the unstructured nature of white noise without the low frequency textures. Consider the problem of rendering a fixed gray level, *g*, with binary pixels whose vertical and horizontal pixel period, or separation, is *S*. The goal is to distribute the binary pixels as homogeneously as possible. These pixels would be separated by an average distance in two dimensions. This distance is called the principal wavelength and for this square pixel case would have the value

$$l_g = \begin{cases} S/\sqrt{g} & g \le 1/2 \\ S/\sqrt{1-g} & g > 1/2 \end{cases}.$$

Since the distribution is assumed to be homogeneous, the corresponding power spectrum would be radially symmetric. The principal wavelength would be manifested as a principal frequency, $f_g = 1/l_g$.

Figure 12. Spectra Characteristics of a blue-noise dither pattern. (a) Energy peak at principal frequency, (b) Sharp low frequency cut-off, and (c) high frequency white noise.

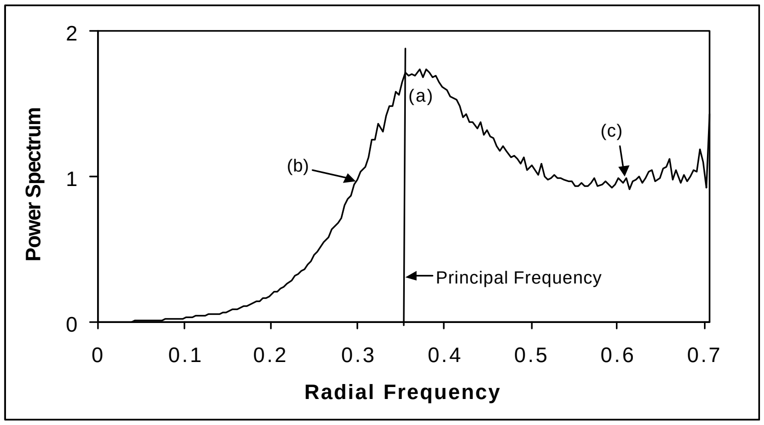

Figure 12 exemplifies the radially averaged power spectrum of a well-formed blue noise dither pattern for a fixed gray level. There are three important features. The pattern should consist of an isotropic field of binary pixels with an average separation of l_g. This corresponds to a peak of energy at the principal frequency (a). The average separation should vary in an uncorrelated white-noise-like manner, but unlike white noise the wavelengths of this variation must not be significantly longer than l_g. So other key features of a blue noise spectrum are (b) the sharp cut-off below the principal frequency, and (c) a flat white-noise-like spectrum above the principle frequency.

5.1 Neighborhood Processes

In image processing, a point process is an operation that uses as its only input the value of the current pixel; a neighborhood process uses as input the value of the current pixel along with values of pixels surrounding it. One neighborhood process that attempts to generate blue noise is the error diffusion algorithm. A graphical illustration of the algorithm is shown in Figure 13(a). Assuming that the input signal varies from g=0.0 (black) to g=1.0 (white) the Threshold block simple sets the output to 0 for values less than 1/2 and to 1 for values greater than or equal to 1/2. The binary output signal is subtracted from the pre-threshold signal to form an error. This error is "diffused" into yet to be considered input values as governed by the Error Filter. The signal consisting of past error values is passed through this filter to produce a correction factor to be added to future input values.

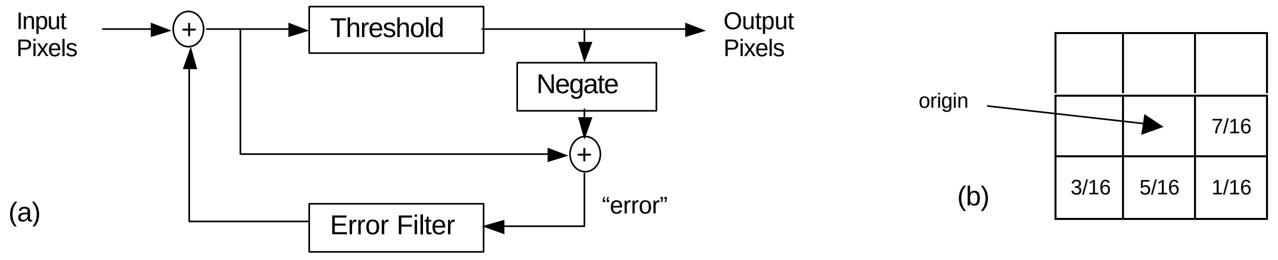

Figure 13. (a) The error diffusion algorithm. (b) Error filter identified by Floyd and Steinberg.

This algorithm was first introduced by Floyd and Steinberg [9], who also proposed the error filter shown in Figure 13(b). The algorithm processes pixels in a raster order, so the only nonzero filter elements are those in front of and below the current pixel. As with all error filters, the elements must sum to one. An image dithered by error diffusion with this filter is shown in Figure 14.

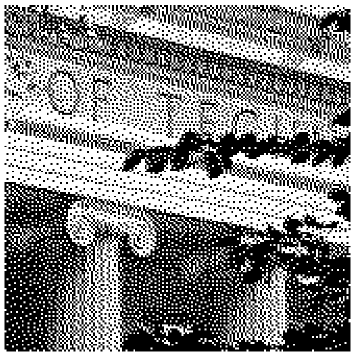

Figure 14. Error diffusion dithering using the error filter identified by Floyd and Steinberg.

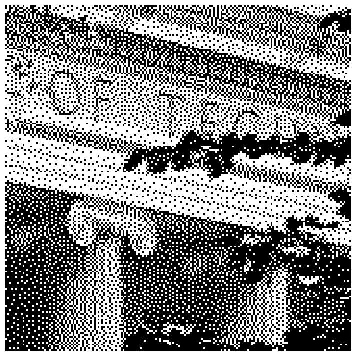

Figure 15. Improved error diffusion using a serpentine raster and a "noisy" weight filter.

Several larger error filters have been proposed that appear to create better looking images. However upon inspection, they look better because the filters tends to sharpen more; areas of flat gray are in fact less homogeneous than the original 4-element filter. Using error diffusion as a sharpening means is less desirable than employing an independent pre-sharpening step because there is no control over the degree of sharpening. The effective sharpening filter intrinsic to the error diffusion algorithm has been measured [10]; besides being hard to control it is also found to be very asymmetric.

Many of the gray level patterns that result from the error diffusion algorithm with the Floyd and Steinberg error filter suffer from directional "worm" artifacts. Also disturbing distortions occur for other patterns, particularly those close to the perfect checkerboard at 50% gray. These anisotropic features are reflected in the radially averaged power spectrum. These shortcomings can be overcome by incorporating two changes in the traditional error diffusion algorithm. First, processing the input on a serpentine raster can break up most of the directional artifacts, and second, adding some random perturbations to the weights in the error filter can further break up stable structures that would otherwise be manifested as unwanted texture.

The result of this modified error diffusion is shown in Figure 15. Gray levels for this process exhibit well behaved blue noise properties, as are plotted in Figure 16. Each of the plots has a peak at the principal frequency for the respective gray level. As gray level increases from 0.03, to 0.50, the distribution of spectral energy moves like a wave crashing into the high frequency wall, then bounces back in the same way above 0.50. The plots for gray levels *g* above 0.50 are very similar to those for $(1 - g)$ but not explicitly drawn to avoid graphic confusion.

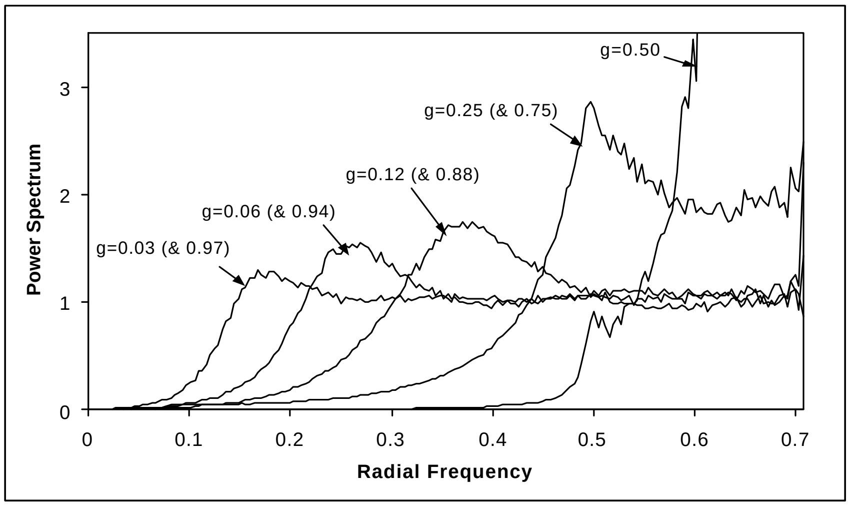

Figure 16. Radial Spectra for various gray levels of for the blue-noise process used to generate Figure 15.

5.2 Point Processes

Blue noise dithering can also be achieved with the point process of ordered dither. The trick of course is using an appropriate dither array. Because of the implementation advantages of ordered dither over neighborhood processes, this has become an active area of research. In the printing industry, ordered dither arrays used for this purpose are often referred to as "stochastic screens".

An overview of approaches to generating blue-noise dither templates is presented in [11]. One approach would be to build an template by directly shaping the spectrum of binary patterns by an iterative process so as to force blue noise characteristics [12].

A very straightforward and effective approach to generating relatively small dither templates of this type is the Void-and-Cluster algorithm [13], and will be outlined here. As with all ordered dither, the array and resulting binary patterns are periodic. Figure 17 illustrates this periodicity. This algorithm looks for voids and clusters in prototype binary patterns by applying a void- or cluster-finding filter at the area under consideration. Because of this implied periodicity, a filter extent will effectively wrap around as shown.

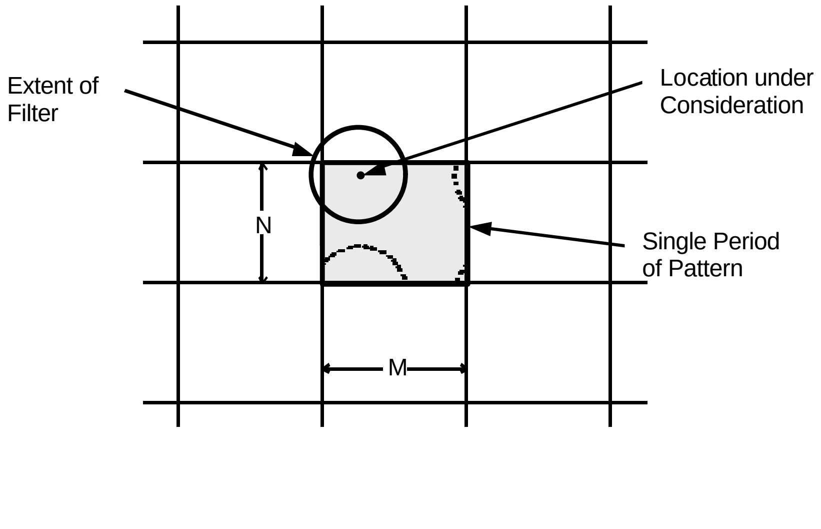

Figure 17. Two-dimensional periodicity of ordered dither pattern, and wrap-around property of void-and-cluster-finding filters.

Binary image patterns are made up of pixels with one of two states that can be thought of as either “1” or “0”, or “black” or “white”. Except for the special case where there is exactly the same number of pixels of each state, there will always be more pixels of one state (“majority pixels”) than the other (“minority pixels”) in any given binary pattern. A void-finding filter considers the neighborhood around every majority pixel in a prototype binary pattern, and a cluster-finding filter considers the neighborhood around every minority pixel. The algorithm uses these filters to identify the biggest void or tightest cluster in the pattern.

We start by relaxing an arbitrary initial pattern to form one that is homogeneously distributed. In Figure 18(a) a 16x16 binary pattern is shown with 26 minority pixels randomly positioned. The purpose of the algorithm is to move minority pixels form tight clusters into large voids. With each iteration the voids should be smaller and the clusters looser. This is done one pixel move at a time until both the voids and stop getting smaller, and the clusters stop getting looser. It turns out that the condition of convergence is quite simple; processing is complete when removing the pixel from the tightest cluster creates the largest void.

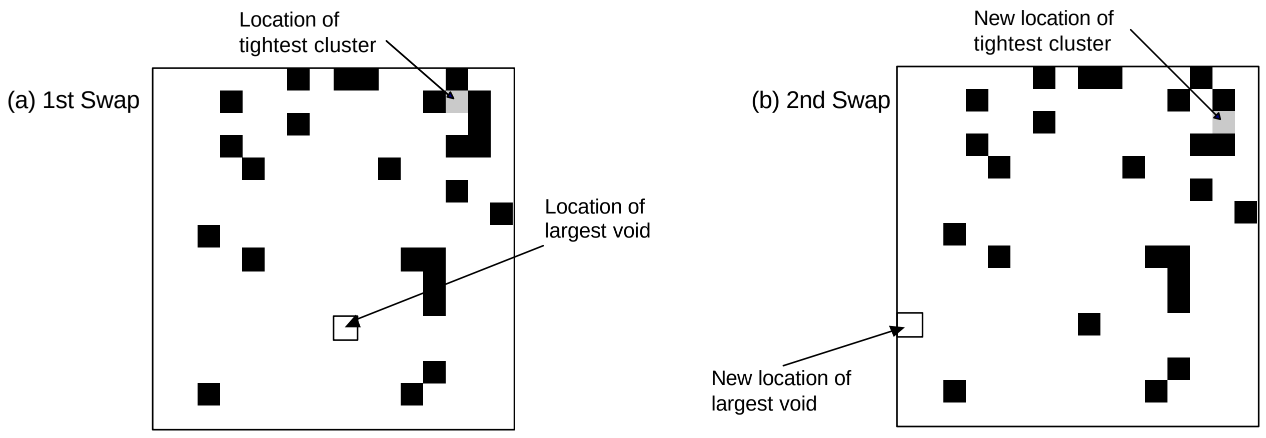
Location of tightest cluster
(a) 1st Swap
Location of largest void
New location of tightest cluster
(b) 2nd Swap
New location of largest void

Figure 18. Example of the first two iterations of the initial binary pattern generator. The 16x16 input pattern has 26 minority pixels.

The minority pixel in the tightest cluster, and the majority pixel in the largest void are identified in Figure 18(a). After the first iteration, the minority pixel in the tightest cluster is moved to the largest void, resulting in the pattern shown in Figure 18(b). Once again, the locations of the new tightest cluster and new largest void are identified. It should be noted that it is entirely possible for minority pixels to be moved more than once; the search for voids and clusters at each iteration is independent of past moves.

The results of this example are summarized in Figure 19 where 12 iterations were needed before convergence. Four periods are shown of both the (a) input pattern, and (b) the relaxed or rearranged pattern to illustrate the wrap-around or edge-abutting consequences of tiling two-space with such patterns. Note how homogeneously distributed the resulting pattern is.

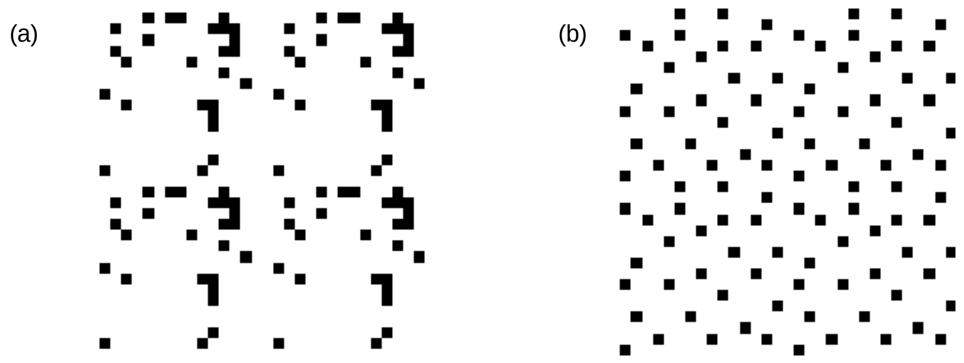

Figure 19. Result of the initial binary pattern generator. Four periods of the 16x16 input pattern (a) and the rearranged pattern (b) are shown to illustrate the wrap-around properties. The input pattern is the same as that shown in Figure 18.

Next, starting with this relaxed pattern as a starting point, a dither template is ordered in parallel. Elements of increasing value in the dither template are entered as minority pixels are inserted into the voids. Then returning to this starting pattern, elements of decreasing value are entered as minority pixels are removed from the tightest clusters.

Figure 20 shows the result of dithering an image with a 32x32 void-and-cluster generated dither array. It should be noted that the image does not appear as sharp as those produced by error diffusion. As mentioned earlier, the added run-time complexity of error diffusion does afford the side benefit of serving as a sharpening filter, even if uncontrollable. As will be shown in the next section, a prefilter as part of a rendering system can make up for this.

It is interesting to note that if a completely empty (all white) pattern is used as a starting point, this algorithm will generate recursive tessellation dither templates. This will in fact also result if the starting point is any of the recursive tessellation patterns.

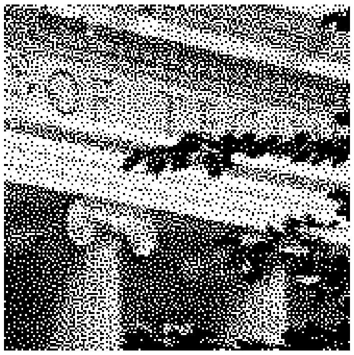

Figure 20. Dithering with a 32x32 void-and-cluster array.

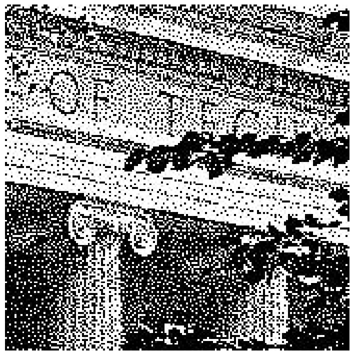

Figure 21. Result of presharpening the same input image used in Figure 20 with a β-2.0 Laplacian, prior to dithering.

6. RENDERING SYSTEMS

The goal of an image rendering system, of which halftoning is a part, is to take device independent image date and tailor it to a target display. Figure 22 illustrates the major phases of an image rendering system: (1) filter and scale, (2) color adjust, (3) dither, and (4) color space convert.

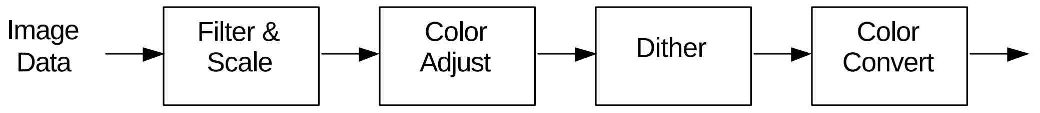

Figure 22. Stages of an image rendering system.

In the first stage, the original image data must be resampled to match the target window or page size. Scaling should be independent in each dimension to allow for asymmetric pixel aspect ratios in either the source data or the target display. A band-limiting filter should be used for reductions and an interpolating filter should be used for enlargements.

Sharpening can also occur in this stage. A typical sharpening scheme can be expressed by the following equation:

$$I_{sharp}[x,y] = I\,[x,y] - \beta\Psi[x,y] * I[x,y]$$

where $I[x,y]$ is the input image, $\Psi[x,y]$ is a digital Laplacian filter, and "*" is the convolution operator. The nonnegative parameter β controls the degree of sharpness, with β=0 indicating no change in sharpness. When enlarging, sharpening should occur before scaling, and when reducing, sharpening should take place after scaling. To illustrate the effect of sharpening, Figure 21 shows an image that was presharpened with a sharpening factor of β=2.0, then dithered using the same process as that of Figure 20.

The second stage of rendering is color adjust, most easily achieved with a look-up table (LUT). In the case of color images, each color component can use a separate adjust LUT. In the case of luminance-chrominance color, an adjust LUT for the luminance component controls contrast and brightness, and LUTs for the chrominance components control saturation.

6.1 Multilevel Dithering

The third stage of Figure 22 is Dithering, the focus of this article. While hard copy display products are capable of marking the color component of a pixel as either on or off, video products generate many levels. When an ordered dither array is used, such as that generated by the method of recursive tessellation, void-and-cluster, or other methods, multilevel dithering can be implemented very efficiently. This section details a means for doing this with minimum hardware or software, yet guarantees output that preserves the mean of the input. It allows dithering from any number of input levels N_i to any number of output levels N_o, provided $N_i \geq N_o$. Note that N_i and N_o are not restricted to be powers of two.

Each color component is treated as an independent image. The input image I_i can have integer values between 0 and (N_i – 1), and the output image I_o can have integer values between 0 and (N_o – 1). A deterministic dither array of size MxN is used. To simplify addressing of this array M and N should each be a power of two. A dither template (such as that in Figure 10) defines the order in which dither values are array are arranged. The elements of the dither Template T have integer values between 0 and (N_t – 1), where N_t is the number of template levels, which represent the levels against which image input values are compared to determine their mapping to the output values. The dither template is central to determining the nature of the resulting dither patterns.

Figure 23 shows a dithering system that comprises two memories and an adder. The system takes an input level I_i at image location $[x,y]$ and produces output level I_o at the corresponding location in the dithered output image. The dither array is addressed by x' and y', which represent the low-order bits of the image address. The selected dither value $d[x',y']$ is added to the input level to produce the sum s. This sum is then quantized by addressing the quantizer LUT to produce the output level I_o.

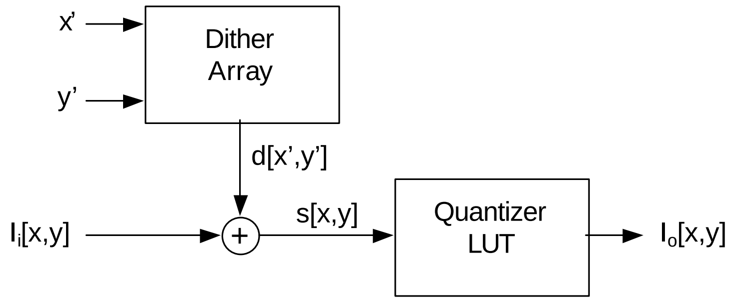

Figure 23. Dithering system with 2 LUTs.

The trick to achieving mean-preserving dithering is to properly generate the LUT values. The dither array is a normalized version of the dither template specified as follows:

$$d[x',y'] = \text{int}\{\Delta_d(T[x',y'] + 1/2\ )\},$$

where int{} is integer truncation, Δ_d, the step size between normalized dither values, is defined as $\Delta_d = \Delta_Q/N_t$ and Δ_Q is the quantizer step size

$$\Delta_Q = (N_i - 1) / (N_o - 1)\ .$$

Note that Δ_Q also defines the range of dither values. The quantizer LUT is a uniform quantizer with N_o equal steps of size Δ_Q.

Using the above expressions, it is possible to simplify the system by exchanging one degree of freedom for another. A bit shifter can replace the quantizer LUT at the expense of forcing the number of input levels N_i, to be set by the system, at least internally. Here we differentiate between input to the dithering system and "real" or "raw" input levels, I_r. If an adjust LUT is used to modify the image data, including a gain makes a "modified adjust LUT". Figure 24 depicts such a system when I_r is the raw input level which can have integer values between 0 and (N_r – 1), where N_r is the number of raw input levels, typically 256. Therefore, the modified adjust LUT must impart a gain of $(N_i - 1) / (N_r - 1)$. The value of f N_i has yet to be found.

The quantizer LUT can be replaced by a simple R-bit right-shifter if the variable Δ_Q can be forced to be an exact power of two: $\Delta_Q = 2^R$. Using the fact that $\Delta_Q = (N_i - 1) / (N_o - 1)$, N_i can be set by the expression

$$N_i = (N_o - 1)2^R + 1.$$

and the value of R can be shown [14] to be

$$R = \text{int}\left\{\log_2\left(\frac{2^b - 1}{N_o - 1}\right)\right\},$$

where b specifies the number of bits with which the input levels I_i are to be represented.

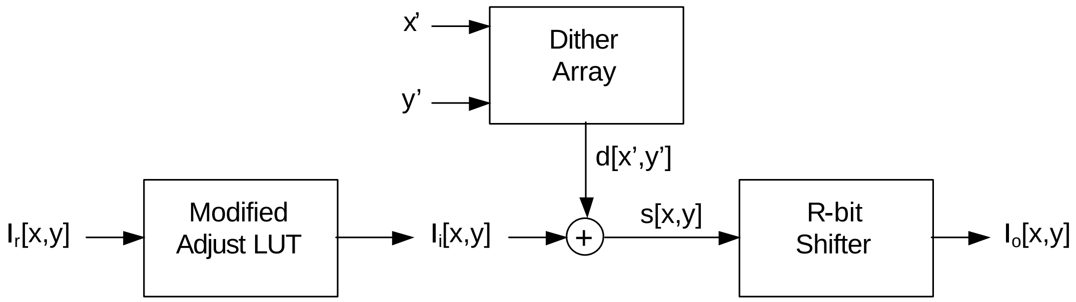

Figure 24. One memory dithering system with adjust LUT and bit shifter.

As an example, consider the case where N_o equals 87 (levels), b equals 9 (bits), N_t equals 1,024 (levels, for a 32-by-32 template), and N_r equals 256 (levels). Thus, R equals 2, meaning that the R-bit shifter drops the least-significant 2 bits. N_i equals 345 (levels); the dither array is normalized by $d[x',y'] = \text{int}\{\Delta_d(T[x',y'] + 1/2)\}$ with $\Delta_d = 1/256$; and the gain factor to be included in the modified adjust LUT is 344/255. This data is loaded into the system represented by Figure 24 and uniformly maps input pixels across the 87 true output levels, giving the illusion of 256 levels.

6.1.1 Bitonal Dithering

Another important simplifying example is the hard copy case when N_o is 2. Consider using the system of Figure 25 with no adjust LUT. Since there would be only one quantization level, the adder and quantizer LUT could be replaced by a comparator as shown in Figure 25. Here the dither array is further normalized to incorporate the quantizer threshold:

$$c[x',y'] = (N_i - 1) - \text{int}\{\Delta_d(T[x',y'] + 1/2)\}$$

and the comparator outputs a "1" when the condition $I_i \geq c[x',y']$ and 0 otherwise. By way of example, suppose a small dither template with N_t = 16 levels is used, and N_i = 256 input levels. Δ_d would equal 255/16, and the system of Figure 25 would yield a perfectly uniform, macroscopically mean-preserved representation of the input.

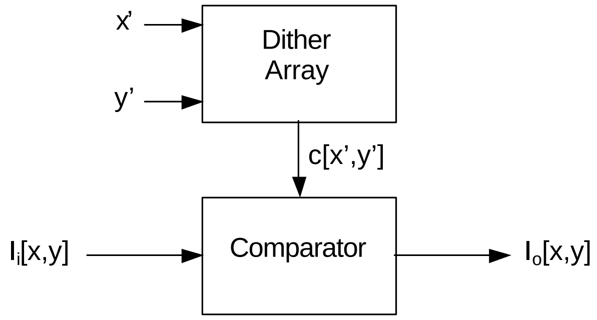

Figure 25. Bitonal dithering system using a comparator.

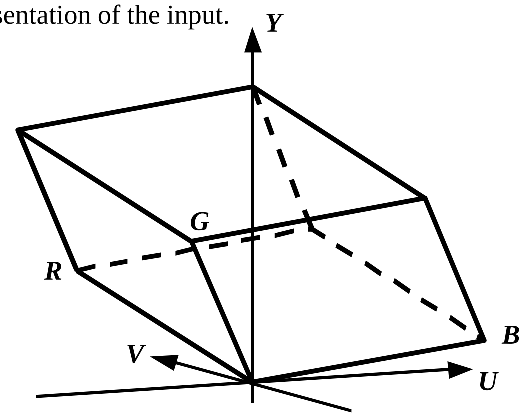

Figure 26. Feasible RGB parallelepiped in YUV space.

6.2 Color Conversion

Referring once again to Figure 22, consider the final rendering subsystem –color-space convert. In the case of video rendering, a frame buffer that is expecting RGB data will not need to convert the color space if the source data is already represented in RGB, as is the case of graphics generation systems. However, uncompressed motion video is essentially always transmitted and stored in a luminance-chrominance space. Although the chomaticities of the RGB primaries of the major video standards vary slightly, the luminance-chrominance space is always YUV. Y represents the achromatic component that is loosely called luminance. U and V are the chrominance components.

Figure 26 shows the parallelepiped of "feasible" RGB triplets in the YUV coordinate system. Feasible RGB points are those that are nonnegative and are not greater than the maximum supported value. RGB and YUV values are linearly related and can be inter-converted by means of a 3-by-3 matrix multiply.

Figure 27 shows the back end of a rendering system that dithers Y, U, and V color components prior to color space conversion. A serendipitous consequence of dithering is that color space conversion can be achieved by means of table look-up. The collective address formed by the dithered Y, U, can V values is small enough to require a reasonably sized color mapping LUT. There are tow advantages to this approach. First, a costly dematrixing operation is not required, and second, infeasible RGB values can be intelligently mapped [15] back to feasible space off-line during the generation of the color mapping LUT.

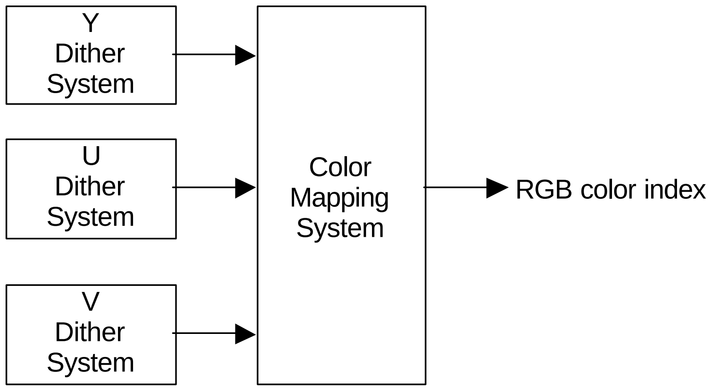

Figure 27. System for dithering three color components and color mapping the collective result.

This second advantage is an important one, because 77 percent of the valid YUV coordinates are in invalid RGB space, that is, in the space around the RGB parallelepiped in Figure 26. Color adjustments such as increasing the brightness or saturation can push otherwise valid RGB values into infeasible space. In traditional systems that perform color conversion by dematrixing, out-of-bounds RGB values are simply truncated; this can change the color in an undesirable way.

7. REFERENCES

[1] K. Knox, "Evolution of Error Diffusion", Proc. SPIE, vol. 3648, pp 448-458, 1999.
also: K. Knox, "Evolution of Error Diffusion", *J. Electonic Imaging*, vol. 8, no 4, pp 422-429, 1999.

[2] R. Eschbach, ed., *Recent Progress in Digital Halftoning II*, Springfield, VA: Society for Imaging Science and Technology (IS&T), 1999.

[3] H. Kang, *Digital Color Halftoning*, Bellingham, WA: SPIE--The International Society for Optical Engineering, 1999.

[4] W.M. Goodall, "Television by pulse code modulation", *Bell Sys. Tech. Journal*, vol. 30, pp. 33-49, 1951.

[5] F.W. Campbell, J.J. Kulikowski, J. Levinson, "The effect of orientation on the visual resolution of gratings", *J. Physiology London*, vol. 187, pp. 427-436, 1966.

[6] R. Ulichney, *Digital Halftoning*, The MIT Press, Cambridge, 1987.

[7] B.E. Bayer, "An optimum method for two level rendition of continuous-tone pictures," *Proc. IEEE Int. Conf. Commun.,Conference Record*, pp. (26-11)-(26-15), 1973.

[8] R. Ulichney, "Dithering with blue noise," *Proc. IEEE*, vol. 76, no. 1, pp. 56-79, 1988.

[9] R.W. Floyd and L. Steinberg, “An adaptive algorithm for spatial grayscale”, *Proc. SID*, vol. 17/2, pp. 75-77, 1976.

[10] Knox, K.T., "Edge Enhancement in error diffusion", *SPSE's 42nd Annual Conf.*, pp. 310-313, May 14-19, 1989.

[11] K. Spaulding, R. Miller, and J. Schildkraut, “Methods for generating blue-noise dither matrices for digital halftoning”, *J. Electronic Imaging*, vol. 6, no. 2, pp. 208-230, 1997.

[12] T. Mitsa and K.J. Parker, “Digital halftoning technique using a blue-noise-mask,” *J. Opt. Soc. Am. A*, vol. 9, pp. 1920-1929, 1992.

[13] R. Ulichney, “The Void-and-Cluster Method for Generating Dither Arrays,” *IS&T/SPIE Symposium on Electronic Imaging Science & Technology*, San Jose, CA, vol. 1913, Feb. 1-5, 1993, pp. 332-343.

[14] R. Ulichney, "Video Rendering", *Digital Technical Journal*, vol. 5, no. 2, pp. 9-18, 1993.

[15] V. Bahl, P. Gauthier, and R. Ulichney,, "Software-only Compression, Rendering, and Playback of Digital Video", Digital Technical Journal, vol. 7, no. 4, 52-75, 1996.