Irrational Halftoning for Electronic Registration

Douglas N. Curry
Xerox, Palo Alto Research Center
Palo Alto, California, USA

Abstract

As xerography moves to intercept offset printing, image quality becomes a key ingredient of success. Classic halftoning methods, which generally deliver good, low noise halftone dots, have fixed positions in the scan field that hinder several possible improvements to these printing systems.

First, exact halftone frequencies and angles would result if dot positions could be adjusted with arbitrary precision. This would improve the design of screen-sets that limit or reduce multiseparation moiré, or allow screen-sets that exhibit the classic rosette structure associated with offset printing.

Second, electronic registration systems could emerge if the halftone dot positions could be adjusted in response to actuation commands from the printer. Such systems would automatically compensate for mechanical distortions caused by bent mirrors, elliptical rollers, and tandem color print stations, for instance, and thus save manufacturing costs for the mechanical system.

Normally, the dot positions are fixed to small integer offsets (the angle corresponds to a “rational tangent”) in the scan field, thus preventing the occurrence of single separation moiré. When fractional dot positions are allowed (irrational tangent), moiré can result. Thus, if the moiré problem can be eliminated for irrational halftoning, the frequency and angle restrictions associated with rational tangent halftoning disappear.

I will present one solution to this problem that sub-samples a halftone cluster function stored in a look-up table to produce reduced moiré separations while printing. Halftone dot locations are computed by hardware, and dot cluster shapes typically do not repeat. I will show a simulation of a three-separation image printed on a 600 spi digital printer that uses irrational offsets (m30°, c75°, and k45°) designed to produce a classic rosette structure. I will also show a simulation of these same dots being electronically registered, or warped.

Introduction

Xerography has several differences from offset printing that can be used to leverage a foothold into the high quality offset market. Since it is evolving from the low end of the market, it is characterized by lower resolution. Lower resolution is both an impediment to high quality and a benefit to high speed printing. If the quality could be improved without increasing the resolution, its speed advantage could be leveraged against offset.

Another major difference is that xerographic prints are made just in time, as opposed to being made in advance, as in offset plates. Just in time printing is burdened with the cost of fast real-time processing, but privileged by the opportunity of variable data per page and thus economical single copies of books and magazines. This could open up a new and potentially lucrative market that offset has been unable to enter.

Fast real time processing and quality at low resolution are therefore key enablers for a xerographic push into traditional and new offset markets. It’s apparent that the former obstacle is becoming moot as processors speed past 1Ghz clock rates, memory prices plummet, and software rips evolve into multi-threaded imaging pipelines with hardware accelerators.

The most difficult problem for xerography to address is therefore the low resolution quality issue. Clearly, offset has the quality advantage with their massive and costly hardware, high speed and well tuned presses that can hold registration to within very close tolerances, using plate sets that are both repeatable and stable over time.

The quality issue for xerography would not be so bad if only one separation were required. Toners, development systems, fast lasers, and image processing have been perfected to offer very good monochrome performance. However, registration problems are encountered when two or more color separations are poorly overlaid.

Registration is important in offset color reproduction, being even more critical than tone reproduction or gray and color balance.[1] Multiseparation misregistration is therefore a more daunting problem for high speed xerographic printers, because the need for speed requires multiple print stations, each with differing registration characteristics. Distortions to images coming from scan line bow, scan non-linearity, process velocity variations, photoreceptor belt thickness variations, job cycling, thermal, and other considerations conspire to degrade quality.

This paper will examine the opportunity of using computation to deform images to match known or measured imperfections of the printer and print stations during printing, thereby canceling registration imperfections between separations. Referred to as *electronic registration*, the cost of moving electrons in images during printing is postulated to be much less than the costs of positioning atoms of massive hardware “iron” with precision manufacturing to achieve perfect cams, mirrors, belts and drums.

There are two image components that make up a single separation, a data layer, and an underlying halftone layer that is the carrier for the data. Both of these layers are based upon regular arrays of data, and both are subject to electronic registration, or “warping”. The same data used to warp one can be used to warp the other because the goal is to warp both by equal amounts. This paper will focus on the more difficult task of warping a halftone grid without incurring too large a moiré penalty.

In order to warp a halftone grid, it is necessary to change the way halftoning is done. Whereas threshold arrays are typically used to make halftones, a different and more general way is presented here that allows the deformation of the halftone grid. This method is called *irrational halftoning*[2].

Threshold Arrays

Traditionally, cluster halftone dot generators have consisted of small, rectangular-shaped memory threshold arrays that are tiled over an image to produce dots of various sizes. The contents of the array is continuously compared with incoming image data, generating halftone dot subpixels for all values of incoming image data that are above or equal to the threshold. The thresholds are arranged inside the array to allow a single halftone dot to grow larger in a predetermined pattern, often clustered, as input data ascends.

Threshold arrays have several advantageous characteristics. They are inherently simple, and require little computation for operation in either hardware of software. They utilize memory efficiently, requiring only a single eight-bit threshold value for each memory element. The array area spans at least one halftone dot and its memory elements are generally accessed in a raster-style pattern, sometimes altered to stack like bricks.

Threshold arrays are very good at efficiently producing rational halftone dots. As the rectangular array tiles an image, the angle of the resulting dot grid is controlled by the tangent of numbers based on the array’s vertical and horizontal dimensions. Clearly, there are a very limited number of these available angles. For instance, if an angle of 30.00° were desired, the arc tangent of four over seven provides an angle of about 29.75°. This is not very close for multiseparation halftoning, and allows no choice at all for the frequency of the dot, which is determined by the length of the hypotenuse of the resulting 4.00 – 7.00 – 8.06 right triangle.

The need to produce halftone dots with greater accuracy in their screen frequency and angle has led to the use of the super cell[3]. This concept allows more possible frequencies and angles by tiling multiple halftone dots side-by-side in a larger threshold array, the larger array size not being an exact multiple of the size of the single-dot array. Thus, each dot of the super cell will be slightly different to accommodate the new and heretofore unobtainable frequency or angle. The need for ever more accurate screens has multiplied the amount of memory required by the supercell array so that the memory-size advantage is compromised, but even then the accuracy required to achieve the exact required frequency and angle is generally deficient. Furthermore, they are just as un-warpable as the single cell arrays.

The presence of a single threshold value in the array presupposes that once a subpixel is turned on, it stays on as the image data ascends. This assumption saves a great deal of memory, but restricts flexibility. For instance, it does not allow for halftone dot modification at midtone, where dot gain performance could be improved by altering the shape of the halftone dot to avoid unnecessary contact with its neighbors. Clearly, multiple thresholds could be provided with appropriate complexity, but this assumes that there is an overriding need to save memory.

Three Dimensional Look-up Table

The method of making halftone dots presented in this paper favors a three dimensional memory array approach. Thresholds and comparisons are eliminated in favor of direct look-up for each image intensity value. Thus, memory size is sacrificed in favor of total output flexibility[4].

The key difference, the one that enables irrational and/or warping performance, however, is that adjacent pixel values on the printed image are not adjacent in the halftone memory as they are in a classic threshold array. Instead, memory accesses span multiple memory locations, and do so with fractional precision.

The contents of the memory must therefore be loaded with a cluster function that presupposes the desired frequency and angle. In essence, many possible halftone dots for a given frequency and angle are superimposed into a single discrete function that is sub-sampled at a constant frequency and angle. Warping is enabled when the frequencies and angles are allowed to vary by small amounts from the constant.

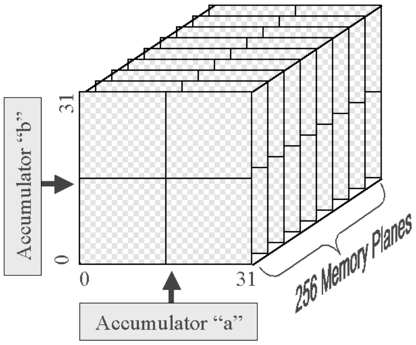

Figure 1. 3-d Halftoner

Consider the halftone generator of Figure 1, that is made up of a square array of memory elements, perhaps 32 on a side. Consider further that there are 256 of these arrays, one for each individual intensity level for a halftone dot.

The contents of each memory element are accessed through two orthogonal address generators called accumulators, which track the a,b coordinates of the samples. At each clock or each new scan, the contents of the accumulators are updated to step through the halftone memory at some frequency and angle. Modulo arithmetic, in this case mod 32, is used to confine the addresses inside the memory during updates, thus tiling the halftone dot over the image.

Figure 2 shows a grid of samples (x's), each row a part of a separate scan generated for that particular halftone dot. Sample locations are quantized to the nearest memory element, but the fractional precision is not discarded.

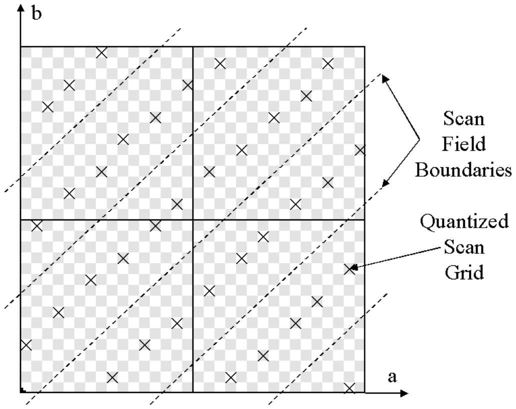

Figure 2. Sample grid & scan boundaries.

The "a" direction is shown along one axis of the halftoner; the "b" direction along the other. This a,b halftone coordinate system is then rotated and scaled with respect to the scanning and data space, identified in Figure 3 as the x,y space. The "x" coordinate is the clock sequence number from the start of scan, and "y" is the scan number counted from the top of the image.

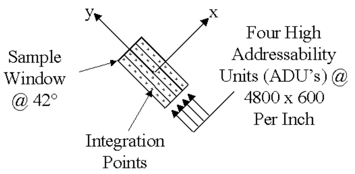

Figure 3. Sample window with 4 adu's.

The halftone dot sample position is therefore a function of x and y. The following equations give the halftone dot sample position in terms of memory address, with arbitrary precision, for each accumulator a and b:

$$a_{x,y} = \mathrm{mod}\left[\left(\frac{F \cdot m}{C}\cos(A)\cdot x + \frac{F \cdot m}{S}\cos(A-90^\circ)\cdot y + P\right), m\right] \quad (1)$$

$$b_{x,y} = \mathrm{mod}\left[\left(\frac{F \cdot m}{C}\sin(A)\cdot x + \frac{F \cdot m}{S}\sin(A-90^\circ)\cdot y + P\right), m\right] \quad (2)$$

F, A and P are the halftone frequency, angle, and phase, respectively, C and S are the clock and scan spatial frequency, respectively, and m is the modulus. For the example presented in this paper:

F = 141.000 dpi
A = 42.000°
P = 0; range 0 to m
C = 1200 memory references per inch
S = 600 scans per inch
m = 32

The accumulators generate a sample grid at a precise frequency and angle while maintaining fractional precision. Hence, an angle of 30 degrees, which corresponds to $1/\sqrt{3}$, is held with as much precision as desired by the number of bits in the accumulators. Generally, the sampling grid is stationary, but for warping the sampling locations can be allowed to vary in small amounts without recomputing the table contents.

Moiré Considerations

Each memory element of the array of 1024 elements contains four bits that represent a set of high addressability units (adu's), in this case four, that sub-divide the fast scan direction into 4800 units per inch. Assume that the output device is at 600 scans per inch, and that the clock is sampling the halftone array at 1200 samples per inch. Figure 3 shows the four adu's, which form a sample window, and their relative size with respect to the halftoner array for a halftone dot with an angle of 42 degrees, and a frequency of 141 dpi. The sample window has been rotated and scaled so that it will exactly tile the plane with no overlaps when centered on the (unquantized) sample grid locations.

Clearly, for irrational angles and frequencies, the center of each individual halftone dot can have any positional relationship whatsoever with the sample grid. In particular, the dot could be shifted in the clock direction x by any amount up to the width of a single addressability unit, or 1/4800 of an inch. After that, the shift repeats. For the x direction this shift is small, and therefore quantization errors to the nearest adu center are not very noticeable in terms of moiré.

In the scan direction y, the situation is very different. The shift only repeats after each scan, or 1/600 of an inch. As the scan field gradually changes its dot phase in the y direction while traversing adjacent halftone dots, the overlap of the dots on the scan field boundaries also changes. Quantization errors of this magnitude are much more noticeable. This uncompensated change in overlap can be a major cause of scan-field moiré, or auto-moiré. Auto moiré can be thought of as the dot grid beating with the scan field, as opposed to multiseparation moiré, where the dot grids beat (overlap) with each other.

Clustering the Halftone Dot Function

Clustering the halftone dot function will partially compensate for auto moiré. Figure 4 shows a possible halftone dot function that has been chosen for its growth pattern that will be used to produce the clustered dot file. The contour plot in the lower left corner shows the function from the point of view of the clustering algorithm. A threshold is set for the clustering algorithm, which generates the shape shown in the next two figures. Points in the a,b field can be determined to be inside or outside the shape by determining if they are above or below the threshold.

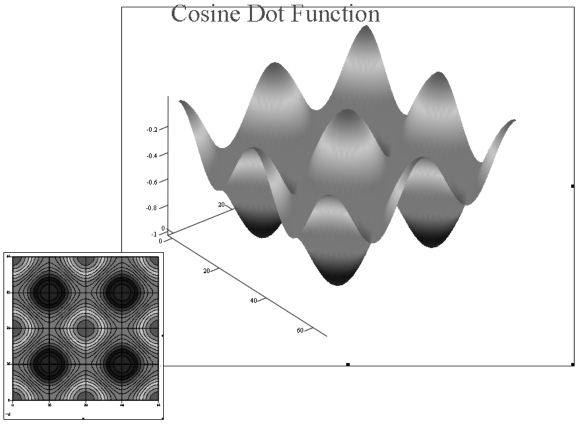

Figure 4.

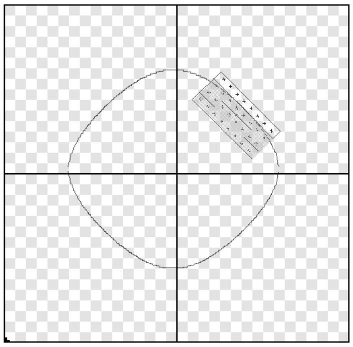

Figure 5. 3 adu's turned on.

The center of the sample window is then positioned at the center of each memory location. Snapping to the center of a memory location is a source of quantization error because irrational sample grid locations will typically never coincide with the memory cell center. This quantization error can be made smaller by enlarging the memory array size beyond 32 x 32. The sample window is repeatedly positioned at the center of each of the 1024 memory locations, and this is repeated for all 256 intensities in the table.

Figure 5 shows a simple case where the sample window of Figure 3 has been positioned over one of the memory locations. The window traverses an edge that is more or less parallel with the long direction of the addressability units. In this case, it is relatively simple to determine how many of the four adu's should be turned on for the current position of the sample window.

An integration within the sample window is performed at the thirty-two interior points. More points can be used to reduce the integration error. The resulting count is used to fill the window one adu at a time, with eight points per adu. In this case, three adu's will be turned on, this result will be stored in the memory location at the center of the sample window.

Figure 6 shows the more difficult case where the sample window (upper right) has enough inside points to turn on one adu, but doing so would not be correct to produce clustered halftones. The problem is thus: As the function is scanned in the x direction, the data must be turned on for just the right amount of time to reproduce the overlap of the halftone dot shape into the current scan field. This overlap area often takes the form of an arc, and can be longer and narrower than the width of the scan field which is 1/600 inch. Since the full width of the scan field is always exposed, the arc area must be integrated and reshaped to form a rectangular cluster of a full scan field width, but with the same area as the overlap. This makes it shorter. Finally, the shortened cluster must be positioned at the center of gravity of the overlap, which can be some distance from the current sample window.

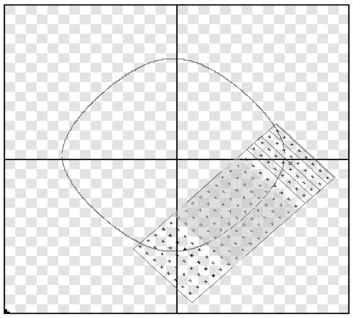

Figure 6. No adu's turned on.

The problem is solved by expanding the sample window until both ends of the arc exits either the top or the bottom of the rectangular integration window, as it also does in Figure 5 without expansion. Integration is performed, and the center of gravity of the arc is determined. The integration count is used to fill up phantom adu's, starting at the center of gravity, working outward, as shown by the dark gray patch. Any intrusion of the dark gray patch into the original sample window at the upper right is recorded. In this case, there is none, so the memory location at the center of the sample window will be loaded with values that indicate all four adu's in the off state. Depending on the path of the arc through the extended integration window, this center fill technique can be supplemented with left fill, right fill or split fill. This process is repeated for each of the 1024 memory locations, creating the cluster function.

Figures 7a, 7b, and 7c show the clustered dot function computed for a particular level. The dot function shown is the same for all 3 phases, and the sampling grid has been adjusted to sample the function at three different y phases. The resulting halftone dot, shown at the right in all three figures, can be read directly by the reader from the points where the sample grid land in the clustered dot function.

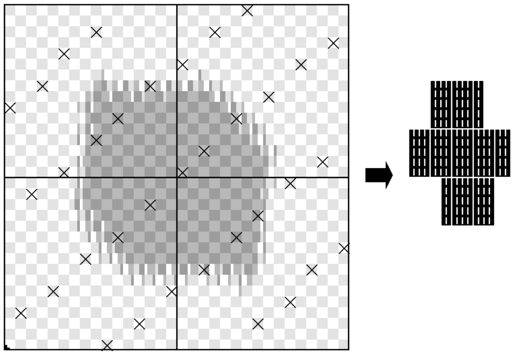

Figure 7a, y Phase is 0

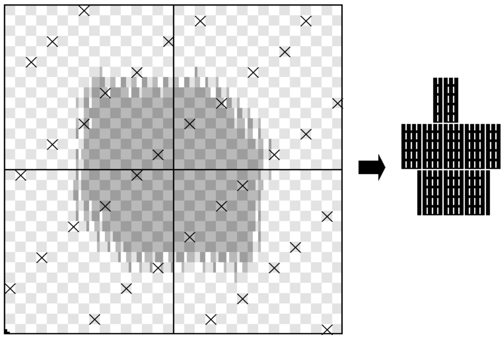

Figure 7b, y Phase is 1/3

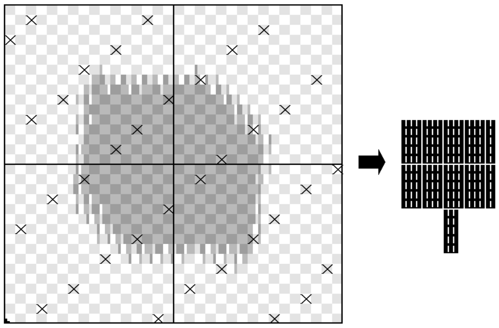

Figure 7c, y Phase is 2/3

Rosettes and Warping

With classic square lattices used for dot screens, even the slightest mismatch in register from color to color will produce enormous moiré.[5] An irrational halftoner is therefore well suited to both produce the correct dot positions for classic halftoning, and also available to continuously adjust the dot positions to achieve multiseparation electronic registration.

The results of being able to produce dots at arbitrary positions in the scan field are shown by example. Figure 8 is a magnified simulation of three separations made at different halftone angles to demonstrate rosette performance. The nominal scan parameters were made similar to the examples given in this paper. The clear centered rosettes can be transformed into dot centered rosettes by setting the accumulator phases to half the modulus at the start of each separation.

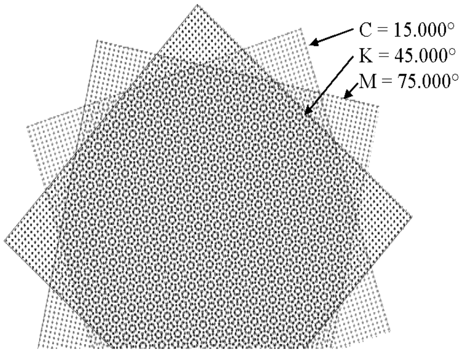

Figure 8. Clear centered rosettes.

Figures 9a and 9b show a simulation of a reference clustered halftone dot grid made with the clustered dot function, and an exaggerated warp made by adjusting the scan grid while printing.

The warp was achieved by applying four separate (and wildly exaggerated!) distortions to simulate electronic registration errors. These warps adjust for scan line non-linearity, scan line bow, process direction skew, and process direction velocity errors, and were made with spline interpolated functions for each separate error, then summed together to be applied to the accumulator increments.[6] Obviously, any real system would use much smaller distortions in the opposite sense: The warped dots would look equally spaced after the distortion compensates for printer spatial errors. Also, in a real system, some of these corrections would be computed in real time from sensors on the machine, as well as using data stored in memory for such corrections as laser scan non-linearity. Note that there is a one-to-one correspondence between the dots in both the reference and the warped image, making them congruent through warping.

Issues

There are some issues revealed by this method of halftoning that deserve consideration. For instance, there is some residual low contrast moiré that can pervade the entire intensity range. This can be explained most easily by considering the highlight part of the dot function (or the empty part in the shadow). This region can be so small that the dot is missed completely at regular intervals by some phases of the sampling grid due to scan grid quantization error. Extending this reasoning into midrange dots, the quantization error will insure that there will be some phase related patterning regardless of dot size. This problem is the focus of current research.

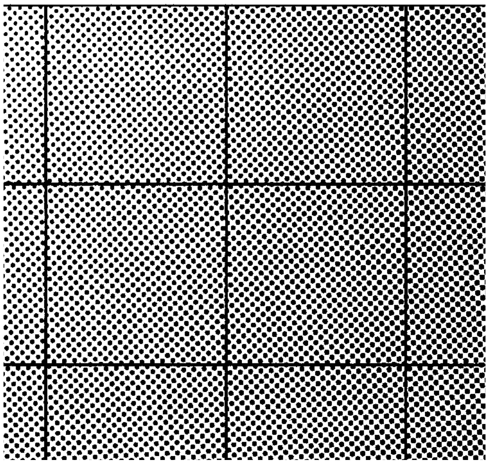

Figure 9a. Reference unwarped regions.

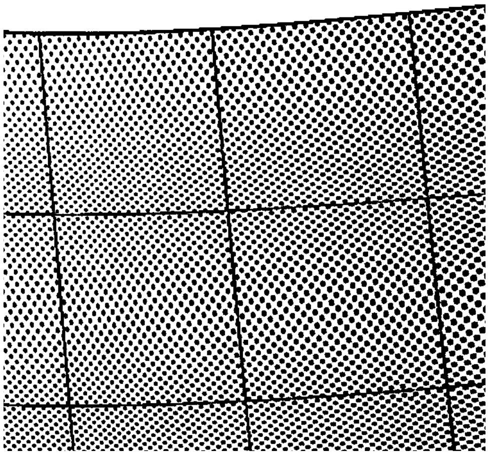

Figure 9b. Regions congruent with 9a through warping.

Another issue is that this method will sometimes produce contouring for dots produced when the frequency and angle is rational. This is easy to see, for rational increments will return to the exact starting point within a small number of steps, skipping accesses to many other memory locations, and thus leaving them unused for the entire print. For the same reason, a particular halftone set achieves better results when it is made with yellow = 0.5°, cyan =15.5°, black = 45.5° and magenta = 75.5°, thus offsetting the angles by a half a degree to avoid the repetitious accesses at 0.0°. As this problem collapses to the same problem encountered in rational halftoning, better attention to growth details should fix it.

Another problem is that the method of integration assumes that the perceived density of the printed image is proportional to the integration area. This assumption is only correct to the first order, and may be inadequate for high quality printing. For example, in a flying spot scanner, the gaussian spot and the xerographic development system will treat an isolated addressability unit on one scan that is attached to a more significant set of adu's on an adjacent scan differently from the same adu simply attached to the end of the more significant set of adu's. Although the area is the same, the response will be different. Although further work is required to compensate for this difference, the problem is by no means restricted to irrational halftoning.

Conclusion

A method for creating a clustering function for irrational halftoning has been shown. The clustering algorithm uses an extended integration window to help reduce moiré by compensating for fractional overlaps of the dot function into the scan field. The resulting cluster function has been shown to produce low-moiré screens at irrational angles, as demonstrated by classic rosette structures. Finally, the irrational halftoning system has been driven with a varying scan grid to demonstrate a functional halftone warping system, suitable for use in electronic registration.

References

1. Michael H. Bruno, *Principles of Color Proofing*, Gama Communications, 1986, pg. 33.
2. David Blatner, et.el., *Real World Scanning And Halftones*, Peachpit Press, 1993, pg. 89.
3. Peter Fink, *Postscript Screening: Adobe Accurate Screens*, Adobe Press, 1992.
4. Douglas Curry, *Halftoning In A Hyperacuity Printer*, USP 5,410,414; April 25, 1995.
5. Isaac Amidror, *The Moire Phenonenon in Color Separation,* Raster Imaging and Digital Typography II, Edited by Robert A. Morris & Jacques Andre, Cambridge University Press, 1991, pg 110.
6. Douglas Curry, *Two Dimensional Linearity and Registration Error Correction In A Hyperacuity Printer,* USP 5,732,162, March 24, 1998

Biography

Douglas Curry received his B.S. degree in Electrical Engineering and Computer Science degree from the University of California at Berkeley in 1996. He has worked for Tektronix in Beaverton Oregon and SRI International in Menlo Park, California. He is currently a Principal Engineer at Xerox's Palo Alto Research Center, in Palo Alto California. He has worked in the Optical Sciences Laboratory, the Electronics and Imaging Laboratory, and is currently in the Computer Science Laboratory. He holds 38 patents related to laser imaging, multi-beam scanning, image enhancement, image security, halftoning, electronic registration, and software imaging processing.