

Hardware Accelerator Component for Image Processing in Industrial Printer Applications

*Marc De Pauw, Koen Vande Velde, Luc Minnebo, Erik Van geel, and Paul A. Delabastita
Agfa-Gevaert N.V.
Mortsel, Belgium*

Abstract

A hardware accelerator component is presented that integrates color correction and an advanced error diffusion algorithm. Because the parameters of the design are flexible, the component can be configured for a wide range of industrial applications where a combination of the highest possible image quality and performance is required.

A picture of a working prototype of the component is shown in Figure 1.



Figure 1. Picture of the hardware component that does color separation and error diffusion

Introduction

Digital Color Printing will Drive Graphic Arts Growth

According to the study made by CAP Ventures Inc. in Ref. 1, the growth for the next five years in the graphic arts industry will mostly be driven by digital color printing. Arguments that support this statement are:

- The strong projected growth of direct advertising;
- The economical pressure on publishers to reduce inventories, leading them to print shorter runs and to print-on-demand;
- The easy integration of digital color printing with digital workflows, in particular Internet based publishing workflows.

Industrial Inkjet Printing

A significant portion of industrial digital color printing is expected to be based on inkjet. One advantage of inkjet printing is that cost for manufacturing ink is intrinsically lower than the cost for manufacturing toner. Another advantage is that the large achievable color gamut and low image graininess make this technology suitable for a wide range of applications, including digital photo finishing.

Advances in micro mechanical engineering technology have allowed for the manufacturing of inkjet heads with more nozzles, higher firing frequencies and multi-droplet size capability, leading in turn to printing systems with extremely high performance. For example, industrial printing systems that produce photographic image quality and print 20 square feet (approximately 2m²) per minute are deemed feasible with current inkjet technology.

Such high printing speeds, of course, demand for image processing functionality with matched performance. This is particularly true for the image processing functionality that has to be performed real time and that is usually embedded inside the printer, such as image decompression, color separation, halftoning and interlacing/shingling.

Molding Advanced Image Processing into Hardware

Recognizing this need has led us to developing a component that performs color separation and error diffusion at a speed of 20.000.000 color pixels per second.

Because image processing has the potential to upgrade the image quality or - equivalently - performance of the design of a printer, we went for image processing algorithms that are considered to be the most advanced in the industry and that, for a given printer design, approximate the "theoretically achievable image quality".

By making the parameters of the design flexible, the component can be configured and customized for a wide variety of printers and applications.

Description of Image Processing Chain

The image processing chain transforms uncorrected color pixels into color corrected and halftoned ink values. We will now discuss in detail the different steps and their specifications.

Color Correction and Separation

The first step in the image processing chain is represented in and consists of the calculation for the color of every input pixel a set of corresponding ink values.

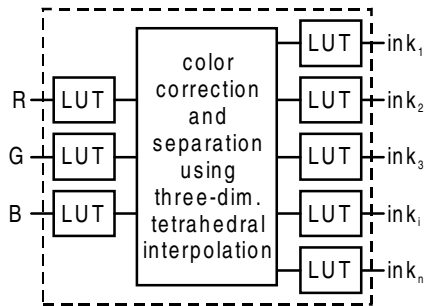


Figure 2. color correction and separation

The input pixels are three-dimensional and can represent, for example, the color of an input pixel represented in sRGB, CIEXYZ, CIELab or in some not-colorimetrically defined three-dimensional space. After a one-dimensional correction of every color component by means of linear look-up-tables in a first step, an interpolation technique in combination with a set of three-dimensional look-up-tables is used to calculate for every input pixel a set of corresponding ink values in a second step. The interpolation technique is of the "tetrahedral" type as represented in Figure 3, and minimizes interpolation error along the neutral and the primary color axes. The contents of the interpolation table are calculated offline by means of a color management system to control, for example, the use of the black printer. The third step of the color correction consists again of a one-dimensional correction of the individual ink values by means of linear look-up-tables.

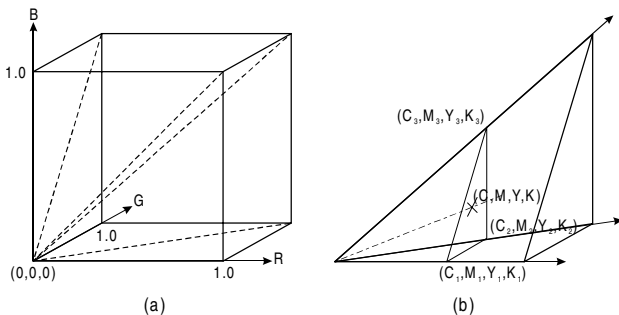


Figure 3. (a) division of RGB input color space into tetraheders and (b) example of interpolation of CMYK ink values within a tetraheder

In a standard configuration, the separation is done into four inks (CMYK). When color separation into more inks is needed, the component allows for two distinct configurations.

In a first - straightforward - configuration, the number of three-dimensional interpolation tables and interpolators is simply expanded to match the number of inks. A maximum of eight inks is supported.

In a second configuration, a set of an ink splitting tables is used that enables to split each of the original four ink values into two or three inks with different densities. The original cyan ink value can, for example, be separated into a light and a dark cyan ink value. Look up tables are provided to control this additional separation step.

We have summarized the most important specifications of the color correction and separation stage in Table 1.

Table 1. Specification of color correction and separation module

Nbr. of dimensions of input color pixel	≤ 3
Nbr. of bits per color component	≤ 16
Nbr. of three-dim. LUT interpolators	≤ 8
Nbr. of entries per dimension in three-dim-LUT	≤ 65
Nbr. of bits per ink channel	≤ 12
Internal precision of image processing	16 bits

Error Diffusion

Because of its potential with regard to optimizing image quality, we have opted for a frequency modulation halftoning based on error diffusion, as opposed to a mask or precalculated bitmap based halftoning technique.

The basis of the algorithm is the four-weighted *Floyd and Steinberg* error diffusion scheme (originally described in Ref. 2). We have added several significant enhancements to overcome some of the well-known problems of the original algorithm.

In the first place *serpentine scan* and *perturbation of the threshold and of the four weights* are supported as described in Ref. 3 by Robert Ulichney.

To support inkjet systems with multiple droplet sizes or inks with multiple densities, the *number of quantization levels is configurable* and can range from 2 to 4095. The improvement in signal to noise ratio by using multi-level over binary halftoning can be appreciated by comparing Figure 4 (a) with Figure 4 (b).

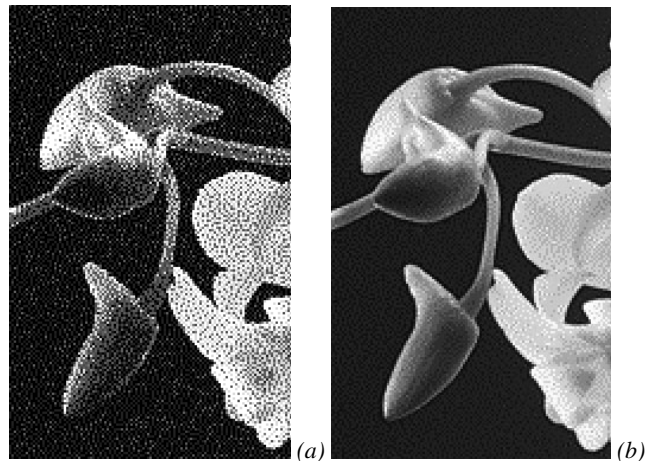


Figure 4. (a) error diffusion with two levels; (b) error diffusion with five levels

The basic Floyd and Steinberg algorithm is essentially a recursive filter, and as such has been known to suffer under specific circumstances from transient or unstable behavior.

Reiner Eschbach (Ref. 4) and Gabriel Marcu (Ref. 5) have suggested improvements for this. The solution of Eschbach consists of adding an additional form of halftone dot phase feedback into the algorithm that suppresses this undesirable transient behavior and improves the homogeneity of the halftone dot distributions for input pixel values near the quantization levels. The technique has been known in the industry under the name of “*imprint function*” and is also implemented as an option in our hardware component. The “strength” of the imprint function is controllable as a function of the input pixel value so that optimal performance under various operating conditions can always be obtained. Comparing Figure 5 (a) with Figure 5 (b) enables to see how the imprint function suppresses unstable transient behavior in error diffusion.

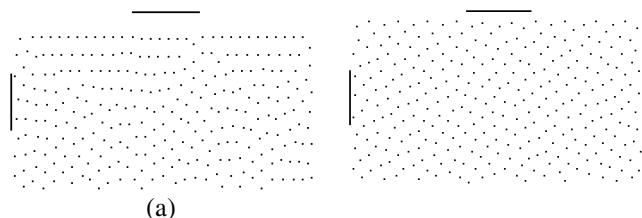


Figure 5. (a) standard error diffusion; (b) error diffusion with dot phase feedback (*imprint function*)

A most significant improvement over the original error diffusion scheme also consists of controlling the relative positions of the halftone dots in the different ink planes to minimize graininess. The solution that we have implemented was earlier presented by Koen Vande Velde at the NIP17 conference (Ref. 6). His algorithm consists of a vector error diffusion scheme in which the quantization of a color into a set of inks is constrained by the output from an additional preprocessing step in such a way that luminance variations - and correspondingly halftoning graininess - are minimized in the final output. The improvement of this technique can be evaluated by comparing Figure 6 (a) with Figure 6 (b).

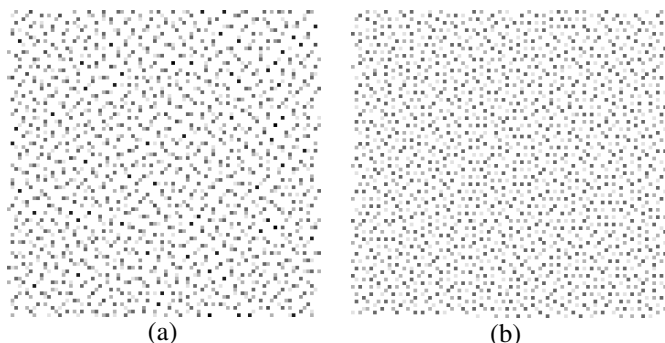


Figure 6. (a) standard error diffusion; (b) error diffusion with constrained correlation to suppress graininess

Selecting the Hardware Technology

Candidate Technologies

The candidate technologies for implementing the image processing are:

- DSP (Digital Signal Processor) technology, represented by Texas Instruments DSP6000 family of products
- FPGA (Field Programmable Gate Array) technology, represented by the Virtex family of products of the company Xilinx
- ASIC (Application Specific Integrated Circuit) technology

Analyzing the Performance Bottleneck

We started off from a highly optimized software version (written in C/C++, the core having been optimized in assembler) of the image processing algorithm. Even on a powerful standard computer platform (Intel Pentium 1.7 GHz) we found that performance of the software code was limited to less than 2 million pixels/sec. Moreover, it was found that switching to even more powerful platforms did not improve performance as much as expected.

An analysis of the architecture of the image processing algorithm reveals that the performance bottleneck is mainly caused by two reasons that are actually interrelated.

A first reason is the sequential nature of software programs running on standard computer platforms. By recognizing that the image processing algorithm itself leaves plenty of room for parallel processing, especially parallel processing of the different color and ink channels, the potential for performance improvement becomes apparent.

A second reason is the processor memory performance gap. While raw computational performance consistently has increased by 60% annually, following Moore's law of rising transistor density, the speed of D-RAM's is increasing only by 7% annually, for reasons that relate directly to the underlying physical principles of these devices. This implies that for image processing, the performance bottleneck is more and more becoming the speed of memory access. Because the performance of processor and memory continue to diverge, technological progress is not going to solve this gap. What is needed is a different computing architecture!

Only avoiding memory access as much as possible can solve this bottleneck. This objective can best be achieved by switching over from a sequential to a pipelined data flow architecture that avoids intermediate storage or results. As it happens, our image processing algorithm, being a cascade of look-up-table operations, interpolation and error diffusion steps lends itself almost naturally to pipelining. Especially in a synergistic combination with parallel processing this approach opens up the way to dramatically improved performance.

Given the above analysis on performance bottlenecks in image processing, we concluded that programmable logic (FPGA or ASIC) offers the best option for performance, because this technology lends itself very well for a design

that exploits both parallelism and pipelining to avoid the memory processor gap.

DSP technology, on the other hand, essentially works its way sequentially through a program and hence suffers from similar memory-processor bottlenecks as standard processors. Parallel processing with DSP technology is not undone, but is far from obvious and certainly expensive, since it involves the deployment of multiple DSP's. The optimized instruction set for digital signal processing does not offset these drawbacks.

Evaluation of Cost

Cost, of course, is an important element when selecting a technology. Cost involves development cost, and fixed and variable manufacturing costs.

The development cost for the three technologies are comparable. The design of both FPGA and ASIC technology is "programmed" in VHDL¹ and the corresponding development costs are therefore comparable. Programming digital signal processors also involves a similar effort and development cost.

For the DSP and FPGA technologies, the fixed manufacturing costs are negligible, while the variable costs are in the range of €100 to €300, depending on the quantities.

ASIC's are a different story in that they require the design and creation of application specific masks. With today's 150 nanometer standard process geometry, this represents an initial investment of approximately €1.000.000. This high initial investment is offset by a low variable cost of €20 or (much) less per ASIC, depending on the quantities manufactured. From this follows that, assuming that the first set of masks is bug-free (which is seldom the case), ASIC technology becomes an economically viable option only for quantities of 10.000 units or more, which is not our case.

As the feature size of integrated circuits continues to decrease over time because of technological progress, the cost of creating masks correspondingly continues to rise, moving the economical turnover point for switching from FPGA to ASIC technology towards even higher volumes. This leaves us with the option of using FPGA's as the preferred technology for our image processing, both in terms of performance and economics.

Image Processing and Interfacing Libraries

An image processing component does not stand by itself, and its design should be viewed in the context of building a system. One of the criteria when selecting a technology is the availability of commercially available support for other image processing functions and standard interfaces.

In the case of the FPGA or ASIC technology, commercial libraries are available that support other image

processing functions such as DCT and wavelet transforms, JPEG and MPEG compression and decompression, and color space converters. As far as interfacing is concerned, support is available for a wide range of standard interfaces including PCI, USB, Ethernet, Bluetooth and Firewire interfaces.

Conclusion

We have presented a hardware component that supports color separation followed by error diffusion for use in inkjet based digital color printers, where the highest possible image quality is required in combination with very high performance.

The component was designed in VHDL and realized using FPGA technology. This technology enables to fully exploit the inherent parallel and pipelined nature of the image processing algorithm and is cost effective for the quantities that are projected. The technology is also commercially well supported when it comes to standard interfaces and additional image processing functionality for building complete systems.

Because the parameters of the component are flexible, a wide range of industrial printing applications is supported.

References

1. Charles A. Pesko, managing director CAP Ventures Inc., "Content on Demand, Reinventing the Printing and Publishing Industry", presented at Print on Demand conference, New York, February 2002.
2. R.W. Floyd and L. Steinberg, "An adaptive algorithm for spatial gray-scale", *Proc. Soc. Inf. Disp.* **17**, 75-77 (1976).
3. R. Ulichney, "Digital Halftoning", MIT Press, Cambridge MA (1987).
4. R. Eschbach, "Error Diffusion algorithm with homogeneous response in highlight and shadow areas", *JEI*, vol. **6**, pp. 288-294 (1997).
5. G. Marcu, "An error diffusion algorithm with output position constraints for homogeneous highlight and shadow dot distribution", *Proc. SPIE* vol. **3300**, pp. 341-352 (1998).
6. K. Vande Velde and P. Delabastita, "Improved Color Error Diffusion", *IS&T's NIP17 Conference Proceedings* pp. 474-476 (2001).

Biography

The five authors work for the image processing group in corporate research of Agfa.

Marc De Pauw is project manager digital design. He has over 20 years experience in developing ASIC's and FPGA's for digital image processing in document scanners and digital printing presses. Marc got his bachelors degree in telecommunication and microelectronics at the Technicum institute in Antwerpen (Belgium).

¹ VHDL: Very high speed integrated circuit Hardware Description Language, a standardized language to describe the design of micro-electronic devices.

Koen Vande Velde holds a PhD in physics. His current activities are directed towards halftoning and color management for inkjet devices and image enhancement technology. Before joining Agfa in 2000, he was working at the Medical Image Computing group of the Catholic University of Louvain (Belgium). He has published several patents in the field of image processing.

Luc Minnebo concentrates on algorithms for modeling image quality of output devices such as inkjet and laser plotters. Before joining Agfa, Luc worked on ISDN software related projects for Alcatel. He got a bachelor's degree in computer science in 1985 from the Industrial Technology College in Antwerpen.

Erik Van geel has been working with Agfa since 1991. He is currently project manager software drivers for the ink-jet devices. Before that he worked for the Digital Printing division as a software developer on the "IntelliStream" front-end for the "ChromaPress" digital printing press. Erik graduated as an Industrial Engineer Electro-Mechanics from the De Nayer college in Mechelen (Belgium).

Paul Delabastita manages the image processing group. He developed the screening technology that is used in the Agfa products. He got his masters degree in electronic and mechanical engineering from the Catholic University in Louvain.