12-7-2001

Document Processing for Automatic Color form Dropout

Andreas E. Savakis

Rochester Institute of Technology

Christopher R. Brown

Microwave Data Systems Inc.

Follow this and additional works at: <http://scholarworks.rit.edu/article>

Recommended Citation

Andreas E. Savakis, Christopher R. Brown, "Document processing for automatic color form dropout", Proc. SPIE 4472, Applications of Digital Image Processing XXIV, (7 December 2001); doi: 10.1117/12.449788; <https://doi.org/10.1117/12.449788>

This Article is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Articles by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.

Document Processing for Automatic Color Form Dropout

Andreas E. Savakis¹ and Chris R. Brown²

¹ Department of Computer Engineering, Rochester Institute of Technology, Rochester, New York 14623

² Microwave Data Systems, Rochester, New York 14620

ABSTRACT

Color dropout refers to the process of converting color form documents to black and white by removing the colors that are part of the blank form and maintaining only the information entered in the form. In this paper, no prior knowledge of the form type is assumed. Color dropout is performed by associating darker non-dropout colors with information that is entered in the form and needs to be preserved. The color dropout filter parameters include the color values of the non-dropout colors, e.g. black and blue, the distance metric, e.g. Euclidian, and the tolerances allowed around these colors. Color dropout is accomplished by converting pixels that have color within the tolerance sphere of the non-dropout colors to black and all others to white. This approach lends itself to high-speed hardware implementation with low memory requirements, such as an FPGA platform. Processing may be performed in RGB or a Luminance-Chrominance space, such as YC_bC_r . The color space transformation from RGB to YC_bC_r involves a matrix multiplication and the dropout filter implementation is similar in both cases. Results for color dropout processing in both RGB and YC_bC_r space are presented.

Keywords: Color dropout, FPGA Implementation

1. INTRODUCTION

Color forms constitute a large number of documents that are scanned using high-speed scanners. In color forms, the information of interest is the text that has been entered, while the document background and document lines, originally placed in the document to facilitate data entry, are not of any practical use. Representative documents of this type are medical forms, insurance forms, census forms, etc. When performing character recognition on these forms, it is desirable to eliminate the color background and lines that are part of the form, and keep only the textual information that is of relevance. Color dropout is the image processing function whose purpose is to convert the scanned color document to a binary image where the form background colors are turned to white and the text colors are turned to black. To accomplish this we need to distinguish between the colors of the background and the colors of the entered text. Color dropout may be viewed as a form of color image rendering, since the image is converted from a full-color form to black and white.

There are several advantages to performing color dropout. First the textual information of interest is enhanced, because it is rendered black, while the background color, that may reduce the text contrast, is suppressed. In addition, the removal of the form lines minimizes interference with the text characters, and may reduce errors during character recognition. Another advantage is that the uncompressed file size is reduced by a factor of 24, since the color image consisting of 24 bits per pixel is converted to a binary image with only one bit per pixel. This fact significantly reduces the storage requirements for the resulting document files.

Color dropout may be accomplished using optical or digital methods. Optical filters have been used when the document form involves a single dropout color. However, optical filters cannot be used with multiple dropout colors, and it is difficult to adjust the optical filter parameters of the optical filters to match nonstandard colors.

Color dropout methods based on digital processing methods sometimes attempt to remove the form lines and background information from the scanned gray scale image by postprocessing. Examples of this approach include [1], where form frames are identified for the purpose of form line removal, and [2], where the distance transformation and its gradient flow are employed to remove form lines. Such approaches may work for specific cases, but require significant computational effort and are very expensive to implement in real-time hardware that are used in high-speed scanners.

Another approach to color dropout, originally developed in the context of optical character recognition, was developed by Rudak [3]. In this work, the average RGB dropout colors in color patches are determined and used in a dropout filter that can be implemented using electronic hardware. The filter bandwidth is adjusted to accommodate for color variations between forms. The advantage of this approach is that the presence of noise, e.g. black specs, does not significantly affect the average color in the color patch considered, and consequently does not affect the final color dropout result. Another approach presented in [4] proposes scanning a blank form, extracting the dropout colors from the blank form, and using them to perform color dropout when scanning other forms.

2. COLOR DROPOUT BASED ON NON-DROPOUT COLORS

The method presented in this paper is designed to operate in a fully automatic environment and is implemented in hardware. The basic assumption is to associate the ink colors with darker colors, such as black and dark blue, and treat lighter colors as part of the document background. During processing, the dark (non-dropout) colors are converted to black, while all other dropout colors are converted to white. Color processing may be done in RGB or Luminance/Chrominance color space [5]. It should be noted that this approach involves some risk in cases where the full range of the non-dropout ink colors is not known. To overcome the risks involved with fully automatic processing, a semi-automatic approach can be adopted where additional dropout and non-dropout colors are interactively specified for particular forms.

2.1 Color dropout using RGB processing

In this implementation of color dropout all of the processing is done in RGB space [5]. The color dropout filter parameters involve the RGB values of the non-dropout colors and the tolerances allowed around these colors. Assuming 8 bits per color channel, the default parameters of the dropout filter is non-dropout colors are black (RGB = 50,50,50) and dark blue (RGB = 50,50,205), and the filter tolerance around these colors is set to R=50 codevalues. Thus, all colors inside the spheres defined by the above centers and radii are the non-dropout colors. Note that the values of 0 and 255 are not used while specifying the non-dropout colors, because they are at the extremes of the data range and are not often encountered in practice. In addition, they do not allow for tolerances in all directions because they are at the endpoints of the intensity range. Placing the sphere centers slightly off the endpoints of the intensity range, allows a wider range of values through the filter using the same tolerance level. To determine whether a color is inside a non-dropout sphere, the distance between the color of the pixel that is processed and all sphere centers should be computed, and then each of the distances should be compared with the associated radius of the non-dropout sphere. In the default setup all of the radii have the same value, but in the future different tolerances can be associated with different colors. If the distance between the pixel of interest and any sphere center is less than the sphere radius, the pixel is classified as belonging to a non-dropout color, and it is turned to black. Otherwise it is turned to white.

The proposed method has several advantages. It is fully automatic and does not require any user intervention. It is very efficient and easy to implement. There is no need for data transformations, since processing is done in RGB space, which is the space of the color data stream. The processing is done on individual pixels without the requirement to examine any neighborhood information. Thus, no part of the image needs to be buffered, and no additional memory requirements are imposed. Along with the advantages mentioned above, RGB processing has shortcomings. The method uses static dropout filters whose parameters are set before the document is

scanned. Thus, the filters are not tuned to the particular colors found in the documents that are processed. In addition, the RGB space is not a uniform color space, and setting the same radius for different non-dropout colors results in perceptually different color differences. Processing in a luminance/chrominance space is discussed next.

2.2 Color dropout using Luminance/Chrominance Processing

This approach improves on the RGB processing method by transforming the RGB image data to a more uniform color space, where color dropout filtering takes place [5]. The RGB color space is the most widely used, but it is device dependent and color differences are not perceptually the same throughout the space. It is possible to transform the RGB values to one of the Luminance/Chrominance color spaces, such as CIE Lab. Here we use the YC_bC_r color space, which consists of Luminance Y , Blue Chrominance C_b , and Red Chrominance C_r . Even though YC_bC_r is not perfectly uniform, it has much better characteristics than RGB and only a matrix multiplication is required for the color space conversion based on the following transformation:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ 0.439 & -0.368 & -0.071 \\ -0.148 & -0.291 & 0.439 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (1)$$

The RGB variables take values in (0-255) and the resulting ranges are (16-235) for Y , and (16-240) for C_b and C_r . The quantities that need to be transformed during the system initialization are the centers of the non-dropout color spheres. During the actual processing of the document, only the RGB values of the pixel under consideration need to be transformed. The decision of whether or not the pixel color is inside a non-dropout sphere is made in luminance/chrominance space by performing comparisons that are similar in nature to those in RGB processing. Since the non-dropout colors are turned black and all other colors white, there is no need to perform the inverse transformation from YC_bC_r to RGB.

Color dropout based on luminance/chrominance processing involves all the steps that are used in RGB processing, as well as one color space transformation from RGB to YC_bC_r color space. This adds some cost in terms of processing time and complexity, but it is well justified, as will be illustrated in the results section. Despite the color space transformation, this approach is quite efficient, since each pixel is processed independently and no part of the image needs to be buffered. However, it remains a static approach, because, as in RGB processing, the non-dropout colors are specified before the image is processed, without any knowledge of the image colors.

3. HARDWARE IMPLEMENTATION

A high-performance image processing core was developed to implement color dropout using Luminance/Chrominance processing. The high-level view of the system architecture is shown in Figure 1. The system is designed to work with a high-speed image scanner running at 120 pages per minute, 600 dpi, and 8.5" x 11" input forms. The system converts 24-bit RGB pixels into black or white binary output. Additionally, system cost is minimized while meeting or exceeding the performance requirements. Given the design parameters, the minimum required system throughput is 67.3 million pixels per second. This appears to be a challenging goal, but this processing rate is actually not difficult to achieve given the nature of the problem. Since the proposed color dropout method is based on point processing without inter-pixel data dependencies, the process lends itself perfectly to pipelining. In fact, due to the extremely large number of pixels per image, the number of pipeline stages is arbitrary. This means that we can choose a large number of pipeline stages in order to run at a very high clock frequency. Due to their performance capabilities, cost effectiveness, and flexibility, an FPGA is an

ideal platform for implementing this image processing engine. As will be discussed in the results section, the design was tested using a Xilinx SpartanXL 50K FPGA.

The first step is to perform the color space conversion as described in equation (1). Our first design step involves the substitution of floating point with integer arithmetic. The fractional values are scaled by a factor of 1024 to facilitate the float to integer conversion. The sign of the numbers is not important at this point since the values are constants. Positive numbers are always added and negative numbers are always subtracted when solving the equation. There is no need to allocate a sign bit for representing these numbers.

The next step in the process is to multiply the 8-bit R, G, and B values with the 10-bit matrix values. The result would normally be an 18-bit value, but this is far more resolution than what is needed. In fact, we only desire a final 8-bit result for each Y , C_b , and C_r value. Therefore, we keep only the 12 most significant bits of each multiply operation. This resolution was chosen in order to ensure that additional roundoff error is not introduced when combining terms into the final Y , C_b , and C_r values. The constant values 16, 128, and 128 have also been appropriately scaled into 12-bit unsigned integer values.

The final color space conversion formula is shown below:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \frac{1}{64} \begin{bmatrix} 263 & 516 & 100 \\ 450 & -377 & -73 \\ -152 & -298 & 450 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} + 16 \cdot \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (2)$$

This is a simple formula to implement since the only required operations are addition, subtraction, and bit shifting. Once the nine initial multiplies are completed, a simple series of unsigned additions and subtractions are required to complete the result. When combining each of the two 12-bit numbers we are interested in only the lowest 12-bits of the result. The carry bit is always discarded. Figure 2 illustrates the color space converter as implemented in the hardware. The first stage performs nine parallel pipelined multiplies for the matrix multiplication. The second stage adds each of the groups of four terms to produce the Y , C_b , and C_r output values. Note that the entire process takes only three clock cycles. The maximum clock speed could be increased by further decomposing this section. However, the tradeoff would be a slightly increased gate count. This is unnecessary since the proposed implementation already meets our speed requirement.

The second step in the algorithm is to compute the distance from the input pixel in YC_bC_r color space to each of the non-dropout colors. Unfortunately the calculation for determining the distance between two points in three-dimensional space is nontrivial, so we approximate this process by simply calculating the distance in each of the three dimensions separately. We can then compare each of the three distances with the threshold. If each of the three distance values is within the threshold, then that pixel qualifies as a non-dropout color. The approximation is such that instead of using a spherical tolerance region around the non-dropout colors, a rectangular region is used.

Sample VHDL code for the distance calculation in the Y dimension is shown below in Figure 3. Of course, the calculations for the C_b and C_r dimensions are exactly the same. The actual hardware as implemented breaks these calculations into two clock cycles: one to compute the signed distance value and another to convert the signed distance to an absolute value. The dropout threshold detector is a simple mechanism that determines whether or not the pixel falls within the threshold of either dropout color. It simply compares the absolute distance from the pixel to the dropout color in each dimension to determine whether or not the pixel falls within the tolerance box around either dropout color. This operation is performed in one clock cycle.

4. RESULTS

The image processor core was implemented in a test system with an architecture as shown in Figure 4. A 50Mhz Motorola 850 CPU running embedded Linux with an NFS file system was used to control the image processing hardware that was programmed into a Xilinx SpartanXL 50K FPGA. A read/write register interface system was developed to allow easy manipulation of the processing parameters, input the pixels, and read the result values.

The intent of the implementation was to get the design running quickly and test the effectiveness and efficiency of the approach in actual hardware. The space usage of the FPGA was at 100%, which was not surprising considering the nine parallel pipelined multipliers contained in the design. The VHDL synthesis tools provided the speed analysis for measuring the maximum clock rate of the system. In this case, the rate was predicted to be 70Mhz, the minimum speed which the design tools were constrained to meet. Unfortunately, due to space limitations, a test bench could not be implemented in the FPGA to exercise the maximum speed of the design.

The hardware performed the processing algorithm very well considering the size constraints of the FPGA. The color space conversion algorithm was tested first, then the overall algorithm results. A Matlab simulation of the algorithm had predicted that the conversion error would be no greater than one in any of the three dimensions, resulting in a negligible difference in output images. The hardware results agreed with the simulation. In either case, the usage of a rectangular versus spherical threshold around the dropout colors resulted in more output error than did the color space conversion. This suggests that some FPGA real estate could be reclaimed simply by using slightly less resolution in the color space conversion process with very little resulting impact on the output image.

As designed, the core provides an excellent amount of processing performance given the FPGA space requirements. However, it also provides excellent system scalability for much greater performance. By using slightly more interface logic, the system can be scaled to much greater performance by widening the data paths and using multiple processing cores. Figure 5 shows an example of such a system. This system uses an input clock rate of 35 MHz with four parallel RGB pixels per clock. The interface doubles the input clock frequency and splits the work between two processing cores, each running at 70 MHz. Of course, there is no need to limit the core frequency. Greater clock frequencies and performance can easily be obtained by simply using a newer technology FPGA.

Representative color dropout results for scanned forms are shown in Figure 6. The color form background consists of red lines, which are suppressed after color dropout processing. In both cases, the results obtained by YC_bC_r processing are superior to the results obtained by RGB processing. These results illustrate that the preferred method for color dropout processing is in the YC_bC_r color space. Future work will involve processing in other uniform color spaces.

REFERENCES

1. B. Yu and A. Jain, "A Generic System for Form Dropout," IEEE Trans. PAMI, 1998
2. J. Mao and K. Mohiuddin, "Form Dropout using Distance Transformation," Proc. ICASSP'95, pp. 328-331, 1995.
3. P. Rudak, "Automatic Detection and Selection of a dropout color using zone calibration in conjunction with optical character recognition of preprinted forms," US Patent 5014329, 1991.
4. Y. Murai and T. Amagai, "Image processing apparatus with function of extracting visual information from region printed in dropout color on sheet," US Patent 5,664,031, 1997.
5. A. Savakis and J. Madigan, "Automatic Color Form Dropout using Luminance/Chrominance Space Processing," U.S. Patent Number 6035058, 2000.

Figure 1. Color Dropout Algorithm Architecture

Figure 2. Color Space Converter Architecture

```

-- Y dimension distance calculation for dropout color 1
Y_distance1 = Y_dropout1 - Y;          -- compute the signed distance
if ( Y_distance1 > 127 ) then          -- convert to an absolute value
    Y_distance1 = 256 - Y_distance1;
end if
-- Y dimension distance calculation for dropout color 2
Y_distance2 = Y_dropout2 - Y;          -- compute the signed distance
if ( Y_distance2 > 127 ) then          -- convert to an absolute value
    Y_distance2 = 256 - Y_distance2;
end if

```

(a) VHDL Code for Distance Calculation

```

-- dropout threshold detector
if ( (Y_distance1 <= threshold and
      Cb_distance1 <= threshold and
      Cr_distance1 <= threshold) or
      (Y_distance2 <= threshold and
      Cb_distance2 <= threshold and
      Cr_distance2 <= threshold ) ) then
    result <= '1';
else
    result <= '0';
end if

```

(b) VHDL Code for Dropout Threshold Detection

Figure 3. Sample VHDL Code for Distance Calculation and Dropout Threshold Detection

Figure 4. Test System Architecture

Figure 5. Ultra-High Performance Color Dropout System (240 pages/min)

(a) Original image detail (shown in gray)

(d) Original image detail (shown in gray)

(b) Color dropout using RGB processing

(e) Color dropout using RGB processing

(c) Color dropout using $YCbCr$ processing

(f) Color dropout using $YCbCr$ processing

Figure 6. Examples of color dropout for standard text (a-c) and low contrast text (d-f)