

DIGITAL SIGNAL PROCESSOR WITH EFFICIENT RGB INTERPOLATION AND HISTOGRAM ACCUMULATION

Hansoo Kim, Joung-Youn Kim, Seung Ho Hwang, In-Cheol Park, and Chong-Min Kyung
Department of Engineering, (KAIST)
373-1, Kusong-dong, Yusong-Gu, Taejon 305-701, Korea

Abstract – In this paper, we present a new digital signal processor developed for digital camcorder applications. Taking the digital image signal from A/D converter, the signal processor generates luminance and chrominance signals of the image using an efficient RGB interpolation algorithm and histogram accumulation. We propose a low-cost RGB interpolation algorithm that has little image degradation and show the usefulness of histogram accumulation implemented in the signal processor.

I. INTRODUCTION

Digital camera applications become so popular that the digital signal processors which extract and process the image signal of CCD (Charge Coupled Device) play an important role in the applications. As the progressive scan applications such as digital still camera and digital television increase, the progressive scan type CCD is being widely used. The progressive scan type CCD has advantages over the conventional interlaced scan CCD [1]. Since the progressive scan CCD is usually based on an RGB primary color filter array, it can provide a better color representation than the interlaced scan CCD that has a complementary color filter array. However, the progressive scan CCD has only one color component, R, G, or B, for each pixel, and thus an RGB interpolation algorithm is essential to calculating three color components for each pixel. In [1], they evaluate several RGB interpolation algorithms with respect to image quality and hardware complexity, and propose an algorithm suitable for the progressive scan CCD, which requires two line memories and complex comparison logic. To cope with the hardware complexity, we propose a new RGB interpolation method that needs only one line memory.

Many digital signal processors include a function to expand the dynamic range of the image [2][3]. For example, to expand the dynamic range, the processor in [2] extracts the variation of the peak level and median level in a frame according to the variation of the target iris control level, and determines the current illumination condition to control iris, gain, and knee circuit. Hence the processor cannot determine the illumination condition until several frames pass. Another

example is the processor in [3] that controls the gamma slope to extract a set of feature data from an input image. As a set of feature data of the image, the processor calculates the numbers of pixels of luminance value in the dark range, in the middle range and in the bright range to make the relationship between the set of feature and the selection of the gamma slope by using a neural network.

In this paper, we propose a new method to determine the illumination condition and expand the dynamic range. The presented processor makes a histogram of the luminance level in a frame and expands the dynamic range by controlling the slopes of knee compensation. By doing the histogram accumulation, the processor can also output histogram-equalized images in real-time. The histogram equalized image is useful in surveillance applications since it is high-contrast even in very dark illumination condition.

This paper is organized as follows. Section 2 describes the functions of the proposed digital signal processor. We propose an efficient RGB interpolation algorithm that can be implemented with the low-hardware cost without image degradation in section 3, and show a histogram accumulation method and its usefulness in section 4. The chip implementation is described in section 5.

II. FUNCTIONS

Fig. 1 shows a block diagram of the proposed digital signal processor. The processor receives the image signal digitized by the A/D converter and performs several image processing to generate luminance (Y) and chrominance (C) signals. The processor consists of eight blocks: digital clamp, RGB interpolation, knee compensation, white balance/RGB to YC matrix, edge enhancement, histogram accumulation, timing generator, and MICOM interface.

Due to system noise, the input image signal from the A/D converter may have a varying black level in each horizontal line. Hence, in each line, the digital clamp block averages values of optical black pixels to determine the black reference level, and calculates the difference between the normal pixel value and the black reference level. In this way, the processor can obtain a pure signal from a noisy input.

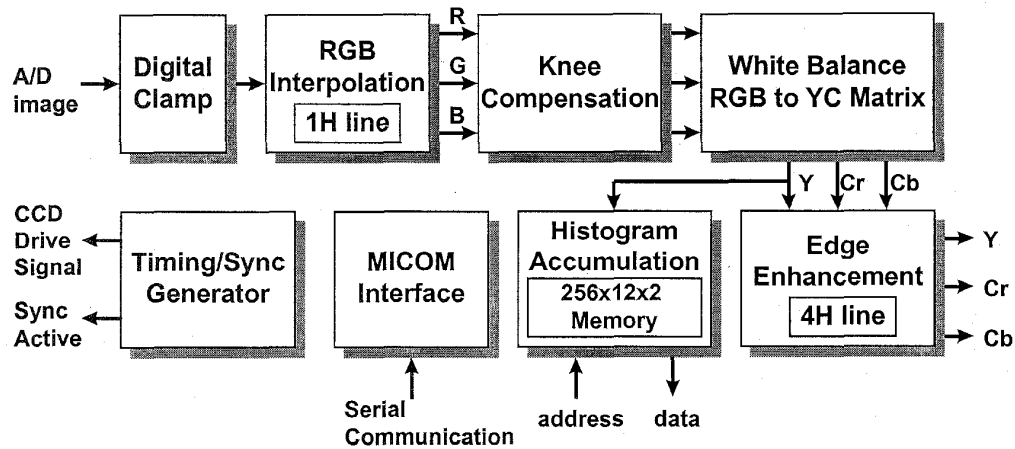


Fig. 1. Block diagram of digital signal processor.

Unlike to three CCD type, the single CCD supports only one color component, R, G, or B, per pixel, because there is one RGB primary color mosaic filter array in the single CCD type. Therefore the RGB interpolation block is required to calculate three color components for each pixel from the neighboring color pixel values. For example, in a B color pixel, the block interpolates its G and R values using the values of the neighboring G and R color pixels, respectively.

The knee compensation block expands the dynamic range of a image. The luminance value of an important object in a image may be in the low or middle range. To expand the dynamic range, the block emphasizes the signal value in the low or middle range and suppresses the signal value in the high range. By doing so, the important object can be represented with the values of wider range. Hence, it is very important to determine adequate slopes and the knee-point with considering the illumination condition. The knee-point is defined as a point at which the applied slope varies.

As the color temperature and the spectral output are dependent on the variation of light sources, the CCD output that is influenced by the light sources cannot represent the true color of the image. To represent the original color, the white balance block must adjust the ratios of the color components conforming to the light sources. And the matrix block converts the R, G and B signals into luminance(Y) and chrominance(C) signals.

The edge enhancement block compensates the edge component of vertical direction as well as that of horizontal direction. For the vertical direction edge compensation, the processor includes four line memories.

In addition, the processor has a histogram accumulation block. The histogram accumulation block

makes a histogram which represents the number of occurrence of the luminance levels in a frame. The histogram information is very valuable to determine a proper knee-point and to perform a histogram equalization. The block also plays an important role in expanding the dynamic range of image according to the light condition, that is, the processor can determine the knee-point and slopes dynamically.

Lastly, the processor includes a timing generator. The timing generator makes a number of timing signals needed to drive a CCD and a CDS(Correlated double sampling) chip. It also generates the sync signals to indicate active pixels and lines, and includes a function to control the speed of electronic shutter.

The blocks mentioned above have many parameters to turn on/off functions and control some coefficients such as matrix, white balance, knee-point, and knee-slope. The parameters are controlled by an external MICOM chip through a serial communication supported by the MICOM serial interface block.

III. RGB INTERPOLATION

Since the progressive scan CCD usually has one RGB primary color filter array, the CCD outputs only one color component per pixel. Therefore, the signal processor has to calculate the other color components using the neighboring color pixel values. In this section, we present an RGB interpolation algorithm and its efficiency.

Fig. 2 shows a typical cell array of the RGB primary color filter type CCD. The bold character in each cell implies the color component and smaller characters the location. The figure shows that the CCD has B and G color

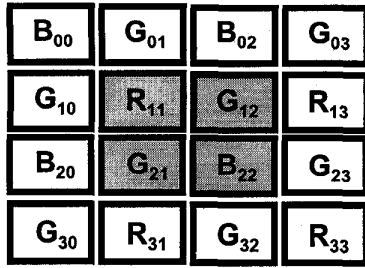


Fig. 2. RGB primary color filter array cells

data alternatively in the even line, and G and R color data alternatively in the odd line. Since the G color data have a major influence on the resolution of an image, the number of G color pixels is twice as that of R or B color pixels in the color filter array, and thus the G color pixel interpolation is different from the R and B color pixel interpolation.

The R and B pixel interpolation algorithm is as follows. Since there are R and B color pixels alternatively in every line, we interpolate the missing R and B pixel values with the pixel values of the current line or the previous line, as shown in equation 1. For example, as there is no B color pixel in the second line, the B_{11} and B_{12} pixels are interpolated with the previous line data, B_{00} and B_{02} . In the third line, on the contrary, there are B color pixels, thus the B_{21} pixel is interpolated with the current line data, B_{20} and B_{22} .

$$\begin{aligned}
 \underline{B}_{11} &= \text{Average}(B_{00}, B_{02}) \\
 \underline{B}_{12} &= B_{02} \\
 \underline{B}_{21} &= \text{Average}(B_{20}, B_{22}) \\
 \underline{R}_{12} &= \text{Average}(R_{11}, R_{13}) \\
 \underline{R}_{21} &= R_{11} \\
 \underline{R}_{22} &= \text{Average}(R_{11}, R_{13})
 \end{aligned} \tag{1}$$

where

$$\begin{aligned}
 \underline{R}_{ab}, \underline{B}_{ab} &: \text{interpolated value of the missing color pixel at } (a, b), \\
 R_{ij}, B_{ij} &: \text{value of color pixel at } (i, j)
 \end{aligned}$$

However, as there are G color pixels in every line, we can interpolate the missing G pixels with three neighboring G pixel values, which are always available with only one line memory. If the processor averages two horizontally neighboring G pixel values to calculate the missing G pixel value, edges may suffer from the blurring effect. In order to preserve the sharp edges, the processor takes a median value among the three G pixel values as the missing G pixel value, as shown in Equation 2.

$$\underline{G}_{11} = \text{Median}(G_{01}, G_{10}, G_{12}) \tag{2}$$

Where

- \underline{G}_{ab} : interpolated value of the missing G color pixel at (a, b) ,
- G_{ij} : value of G color pixel at (i, j)

To show the efficiency of the proposed interpolation algorithm, five RGB interpolation algorithms described below are compared.

- 1) Copy method : copies the previous G pixel value in the horizontal direction
- 2) Average method I : averages two horizontally neighboring G pixel values
- 3) Average method II : averages four horizontally and vertically neighboring G pixel values
- 4) Sort/Average method [1] : averages two median value of four horizontally and vertically neighboring G pixel values
- 5) Proposed method : takes a median value among three neighboring G pixel values

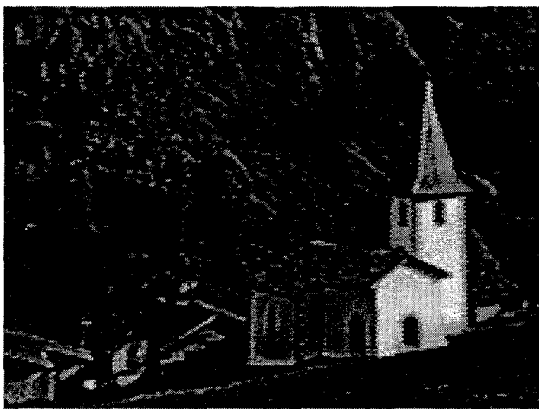
Table 1 shows the hardware complexities of the interpolation algorithms. The Copy method requires no line memory for G interpolation as the previous pixel values are directly used for the missing pixels. Be aware that it still needs a line memory for the R and B interpolation. The Average method I uses two horizontally neighboring pixels, and thus requires a line memory for R and B interpolation and one adder for G interpolation. Since the Average method II averages four neighboring pixels, it requires two line memories and six adders. The Sort/Average method is a variation of the Average method II. Instead of averaging four neighboring pixels, the method averages two median values among four neighboring pixels. Therefore, the method needs two line memories and a sorting engine that is composed of six comparators and a complex selection logic. Finally, the proposed method requires only one line memory, three comparators and a simple selection logic to take a median value among three neighboring pixels. Therefore, in terms of hardware complexity, the Average method II and the Sort/Average method that need two line memories are more expensive than the others.

Fig. 3 shows the quality of images resulted from the interpolation algorithms. In Fig. 3(a), the interpolated image is very coarse in pixel size and is sawtooth, because it only copies the neighbor pixel value. In Fig. 3(b), image degradation occurs near edges, that is, the edge components are very unclear around the roof and wall of the house on the right-hand side. In Fig. 3(c), (d), and (e), the edge component is clear over that of Fig. 3(b) and the

images are not sawtooth. This implies the Average method II, the Sort/Average method and the proposed method are better than the others in terms of image quality. As a result, the proposed method is very efficient in that it can provide good quality images with inexpensive hardware cost.

Algorithm	Hardware complexity
Copy method	1 line memory
Average method I	1 line memory, 2 adders
Average method II	2 line memories, 6 adders
Sort/Average method	2 line memories, 2 adders, 6 comparators
Proposed method	1 line memory, 1 adder, 3 comparators

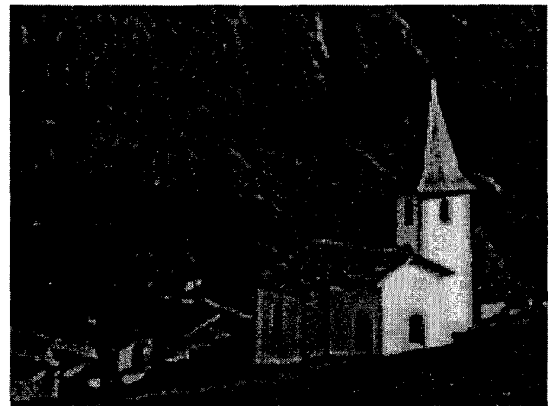
Table 1. Table of comparison among several interpolation algorithms



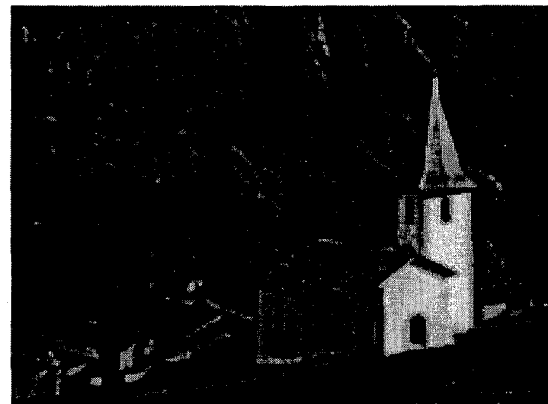
(a) Copy method



(b) Average method I



(c) Average method II



(d) Sort/Average method



(e) Proposed method

Fig. 3. Result images of each interpolation algorithm

IV. HISTOGRAM ACCUMULATION

For expanding the dynamic range of the image, the proposed processor performs the knee compensation as explained in section 2. It is important for an adequate knee compensation to calculate a knee-point and knee-slopes based on the illumination condition. In addition, surveillance applications require high-contrast images regardless of the illumination condition. In this section, we show that the histogram accumulation is very useful for both the knee-point determination and the histogram equalization.

As stated before, a histogram of an image represents the frequencies of the occurrence of the luminance levels in a frame. Hence, the processor can know the illumination condition of the object by investigating the histogram. For example, if the histogram is well distributed to the entire luminance range, its illumination is in front-light condition. On the contrary, if the histogram is distributed to a small range, its illumination is in back-light or excessive-front light condition. In these cases, the images may be low-contrast and indistinguishable.

The histogram information generated in histogram accumulation block is transferred into the MICOM, where a knee-point determination program is running. First, the program searches a peak point in the histogram. Second, the program investigates the histogram around the peak point and determines the illumination condition and searches the x-coordinate of the knee-point of which the histogram value is less than the predetermined value, for instance, about 1% of the histogram value of the peak point. If the x-coordinate of the knee-point is far from the peak point, the image is in the front-light condition. The processor turns off the knee operation for such a high-contrast image. Otherwise, the image requires the knee operation and the program proceeds to the next step. The y-coordinate of the knee-point is determined by the ratio of the summation of the histogram values of the levels less than the x-coordinate of the knee-point and the number of total pixels. From the x-, y- coordinate of the knee-point, knee-slopes are calculated automatically. Then, these parameters are transferred to the signal processor that performs an adequate knee compensation. To expand the dynamic range of the object with high luminance level as well as the object with low luminance level, a multiple-point knee algorithm is being developed.

A histogram equalization transform is a powerful technique for image enhancement[4]. Because of inadequate illumination condition, images of the CCD may have very low-contrast with narrow histogram and may be indistinguishable. If a camera system is used in surveillance applications, it is very important to obtain high-contrast images even under dark environment. The histogram equalization algorithm distributes the luminance level in order to transform the input image into a high-contrast image by extracting the relationship between the input and the output luminance levels. The histogram

information is applied to making a cumulative distribution function of the luminance levels. If we assume that consecutive frames in time are highly correlated and have almost the same histogram feature, the processor can generate high-contrast images that are easily distinguishable by using the previous histogram information to stretch the dynamic range over the entire range.

Fig. 4. shows a block diagram of histogram accumulation. The block is composed of two memories, histogram memory and accumulation memory. The histogram memory contains frequencies of the occurrence of the luminance levels in a frame. When a frame is over, the data in the histogram memory are accumulated in the increasing order of address and are stored in the accumulation memory. As the address counter reaches 255, the accumulation memory contains the cumulative distribution function of the luminance levels, and the block can generate a histogram-equalized image. To make the histogram of the next frame, the histogram memory is initialized to zero. To determine the knee-point, the MICOM reads the contents of accumulation memory by using the *ext_addr* and *histogram_data* signals. Since an entry of the accumulation memory represent a cumulative value, the histogram information is calculated from the difference between consecutive values.

Using the histogram information, the processor can also perform the dynamic knee-point determination and the histogram equalization transform. In conclusion, the histogram accumulation is a very useful feature in CCD applications.

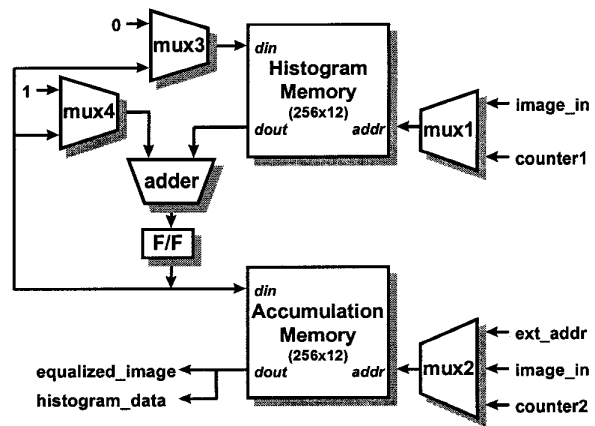


Fig. 4. Block diagram of histogram accumulation

V. VLSI IMPLEMENTATION

The digital signal processor was fabricated using a 0.6 μm double metal CMOS process. Fig. 5 shows a photograph of the digital signal processor containing about

51,000 gates and about 4Kbyte memory in a die size of $9.30 \times 9.29 \text{ mm}^2$. The chip is packaged with 208 TQFP. The characteristic of the digital signal processor is shown in Table 2.

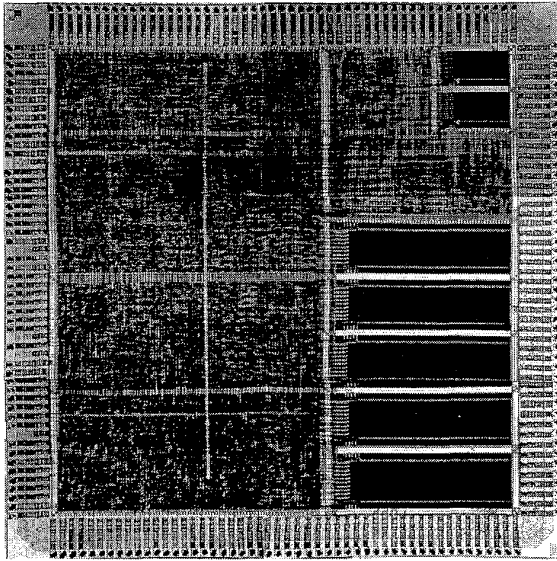


Fig. 5. Photograph of Digital Signal Processor

Gate count	51,000
Memory	4 Kbyte
Die size	$9.3 \times 9.29 \text{ mm}^2$
Input data width	10 bits
Output data width	8 bits
Operating frequency	24.54 MHz
Power consumption	700 mW
Latency	817 cycles
Power supply	5V, 3.3V

Table 2. Characteristic of Digital Signal Processor

VI. CONCLUSION

We present a digital signal processor developed for digital camcorder applications. The processor extracts a image signal from a CCD and performs some image processing. To implement the processor in a small size, an efficient RGB interpolation algorithm is developed. The algorithm that can be implemented with only one line memory has produced almost the same image quality as those that have more hardware complexity. In order to determine the current illumination condition, a histogram accumulation method is presented. By using the histogram accumulation, the processor can control the knee

compensation adaptively and generate histogram equalized images in real time. The feature makes the processor very useful in surveillance applications.

VII. REFERENCES

- [1] Hidemori Zen, et al., "A New Digital Signal Processor for Progressive Scan CCD," IEEE Trans. on Consumer Electronics, Vol. 44, No. 2, May 1998, pp. 289-296.
- [2] Atsushi Morimura, et al., "A Digital Video Camera System," IEEE Trans. on Consumer Electronics, Vol. 36, No. 4, November 1990, pp.866-876.
- [3] Shigeo Sakaue, et al., "Adaptive Gamma Processing of the Video Cameras for the Expansion of the Dynamic Range," IEEE Trans. on Consumer Electronics, Vol. 41, No. 3, August 1995, pp.555-562.
- [4] Anil K. Jain, "Fundamentals of Digital Image Processing," Prentice Hall.

BIOGRAPHY

Hansoo Kim received the B.S. degree in Electronics Engineering from Yonsei University in 1994 and M.S. degree in Electrical Engineering from KAIST in 1996, respectively. Now he is currently working toward Ph.D. degree in the Department of Electrical Engineering in KAIST. His research interests include algorithms and VLSI architectures for video compression, and low power design.

Joung-Youn Kim received the B.S. degree in Electronics Engineering from Yonsei University in 1996. Now he is currently working toward M.S. degree in the Department of Electrical Engineering in KAIST. His research interests include algorithms and VLSI architectures for image signal processing.

Seung Ho Hwang received the B.S. degree in Electronics Engineering from Seoul National University in 1979, the M.S. degree in Electrical Engineering from KAIST in 1981, and the Ph.D. degree in Electrical Engineering from University of California at Berkeley in 1989. In 1990 he joined the faculty of the Department of Electrical Engineering, KAIST where he is now an Associate Professor. His research interests include CAD for VLSI and video signal processing.

In-Cheol Park received the B.S. degree in Electronics Engineering from Seoul National University in 1986, and the M.S. and the Ph.D. degrees in Electrical Engineering from KAIST in 1988 and 1992, respectively. From 1995 to 1996, he worked at IBM T.J.Watson Research Center, Yorktown, New York as a postdoctoral member of the technical staff in the area of circuit design. In 1996 he

joined the faculty of the Department of Electrical Engineering, KAIST where he is now an Assistant Professor. His research interests include CAD algorithms for high-level synthesis and VLSI architectures for general-purpose microprocessors.

Chong-Min Kyung received the B.S. degree in Electronics Engineering from Seoul National University in 1975, and the M.S. and the Ph.D. degrees in Electrical Engineering from KAIST in 1977 and 1981, respectively. From 1981 to 1983, he worked at Bell Telephone Laboratories, Murray Hill, New Jersey as a postdoctoral member of the technical staff in the area of semiconductor device and processor modeling. In 1983 he joined the faculty of the Department of Electrical Engineering, KAIST where he is now a Professor. His research interests include CAD algorithms for VLSI, computer graphics, VLSI architecture for microprocessors and DSP.