Development of a Generic RGB to HSV Hardware

Tatsuya Hamachi*, Hiroyuki Tanabe, Akira Yamawaki

Kyushu Institute of Technology, Fukuoka, Japan (804-8550)

*Corresponding Author. serikawa@elcs.kyutech.ac.jp

Abstract

In a road traffic sign recognition, facial recognition, etc., HSV color space which was suitable for distinction of the color rather than RGB color space is used. However, if small-scale embedded devices perform a HSV conversion process with much computational complexity by software, conversion will take a lot of time. Moreover, although there are those who are performing HSV conversion by hardware, it is the hardware only for research, or, accuracy at the time of fixed-point, working speed, or circuit scale are not clarified. That is, there is no flexibility. It is useful in application to small-scale embedded devices to perform a HSV conversion process by hardware from the field of performance or power-saving. In this paper, a generic RGB to HSV hardware is developed for improvement in the speed of processing, and power-saving. It processes by a fixed point and it is shown that arithmetic precision is equivalent to a floating point. In this time, SPARTAN-6 XC6SLX16-CSG324 (FPGA) of Xilinx is used. And it mounts to FPGA and clarifies processing time, a circuit scale, and clock frequency in a sequential version and pipeline version. As a result, it was shown that there is accuracy equivalent to a floating point, and processing time, a circuit scale, and clock frequency were clarified in a sequential version and pipeline version. Thereby, the validity of HSV conversion hardware in small-scale embedded devices were shown.

Keywords :HSV conversion, FPGA, Verilog HDL.

1. Introduction

In a road traffic sign recognition, facial recognition, etc., HSV color space is used. This is because HSV color space is suitable for distinction of color rather than RGB color space. However, the conversion to HSV from RGB has much computational complexity. Since a lot of time will be taken if this is calculated by software, HSV conversion hardware is developed. This thinks that it is useful from the field of performance or power saving in the application to small-scale embedded devices.

So far, hardware-ization of the HSV color space conversion is carried out [1-3]. By literature [1], HSV color space conversion hardware is included as a part of object detection processing in robot soccer. Similarly, by literature [2], it is included in a part of processing of color image reinforcement, by literature [3], it is included as a part of processing of the performance gain of a color classification. It is not taking into consideration that these previous works are making HSV color space conversion hardware specific processing, and it is used for them general-purpose. So, the interface of HSV color space conversion hardware is not shown. Therefore, it is difficult to take out and use only HSV color space conversion hardware.

Moreover, about performance, only the processing time of the whole processing is shown by literature [1-3], and the processing time of HSV color space conversion hardware is unknown.

Furthermore, in the case of hardware-izing of processing, it is common to be fixed-point-ized for the purpose of the improvement in the speed by reduction of hardware scales or simplification of processing, power-saving, etc. [4]. And the accuracy also becomes a problem when it fixed-point-izes. However, the previous work has not made reference at all about those things.

In addition, as far as an author gets to know, HSV color space conversion hardware does not exist in commercial Intellectual property (IP).

In this research, in order to solve these problems, the generic HSV conversion hardware which clarified the details of hardware is developed. Moreover, since HSV conversion is performed by a fixed point, the accuracy of the HSV conversion result by the fixed point to a floating point and the details of fixed-point-izing will be shown.

DOI: 10.12792/iciae2013.034

And, it mounts to FPGA and clarifies the time which processing with a sequential version and processing with the pipeline version take, a circuit scale, and clock frequency.

2. Color space

2.1 RGB color space

RGB color space (RGB color model) is a kind of the representation scheme of a color. It is a kind of the additive color mixing which mixes three primary colors (Red, Green, Blue) and reproduces a broad color. The RGB color space itself does not define what kind of color "red", "green", and "blue" mean in colorimetry study.

2.2 HSV color space

HSV color space is a color space which consists of three ingredients, Hue, Saturation-Chroma, and Brightness-Lightness-Value.

H is a kind of color and the range is 0-360 degrees.

S is the vividness of color and the range is 0-100%.

V is the brightness of a color and the range is 0-100%.

Moreover, There are some models in a HSV color space. An annular HSV color space model is shown in Fig. 1. A cylindrical HSV color space model is shown in Fig. 2. The HSV color space model of a cone is shown in Fig. 3.

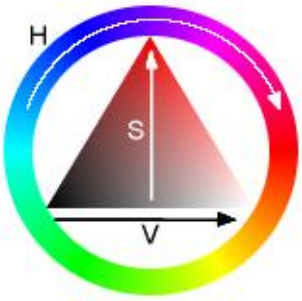

Fig. 1. An annular HSV color space.

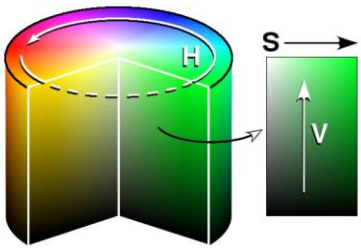

Fig. 2. A cylindrical HSV color space.

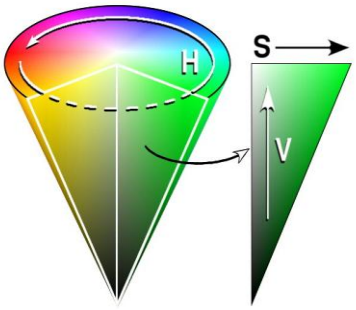

Fig. 3. A cone HSV color space.

2.3 The advantage of the HSV color space to a RGB color space

At a RGB color space, a color is decided by red and a green and blue rate. Since green and a blue ingredient are included also by a color which is judged to be red, distinction of a color is difficult. By changing into a HSV color space, a color can be distinguished out of a certain range. This is the reason a HSV color space is suitable for distinction of a color.

By preparing a threshold value in hue (H), distinction of a color is possible.

2.4 Conversion to HSV from RGB

This time, RGB to HSV conversion is performed using a conical HSV color space. The algorithm of RGB to HSV conversion is shown in literature [5]. The maximum and the minimum are chosen from R, G, and B which were inputted. Then, HSV equivalent to the color defined by RGB can be determined with the following numerical formula. By a formula (1), calculate H, S is calculated by the formula (2), and V is calculated by the formula (3).

$$H = \begin{cases} 60 \times \frac{G-B}{MAX-MIN} + 0 & (if\ MAX = R) \\ 60 \times \frac{B-R}{MAX-MIN} + 120 & (if\ MAX = G) \\ 60 \times \frac{R-G}{MAX-MIN} + 240 & (if\ MAX = B) \end{cases} \quad (1)$$

$$S = MAX - MIN \quad (2)$$

$$V = MAX \quad (3)$$

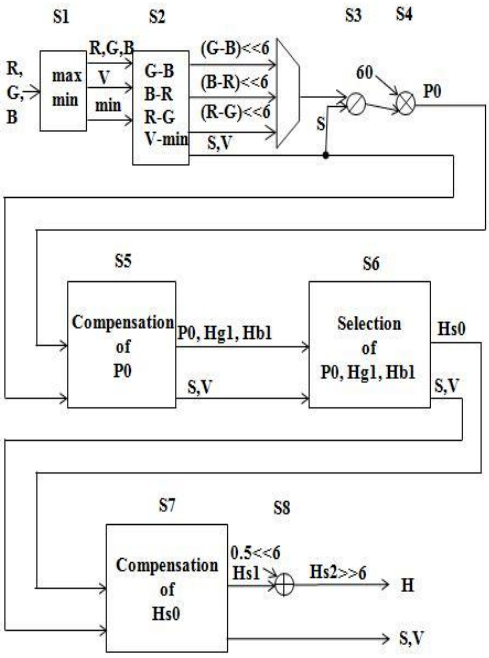

Fig. 4. The data path of hardware.

3. The design of HSV conversion process hardware

The data path of HSV conversion process hardware (conversion HW) is shown in Fig. 4.

In the conversion HW, in the state S1, the maximum and the minimum are chosen from R, G, and B which were inputted, and the maximum becomes the brightness V.

In the state S2, G-B, B-R, R-G, and MAX-MIN (chroma saturation S) are calculated.

In the state S3, division of the three color differences (G-B, B-R, R-G) is done by MAX-MIN (S). Here, it is made the 16-bit fixed point whose 6 bits of low ranks are decimal fractions and whose top 10 bits are integer part. Specifically, color difference is shifted to the 6-bit left and it is divided by MAX-MIN. In addition, since the hue H is undefined in the case of MAX-MIN = 0, it is returned to as -1 here.

In the state S4, 60 is hung on the operation result of S3 (P0, P1, P2).

In the state S5, 120×2^6 is added to P1 and 240×2^6 is added to P2 (Hg1, Hb1).

In the state S6, select P0 when MAX is R, select Hg1 when MAX is G, select Hb1 when MAX is B, and the selected thing is set to Hs0.

In the state S7, when the value of Hs0 is negative, 360×2^6 is added in order to use the value in 0-360 which are the range of the value which H can take (Hs1).

In the state S8, 0.5×2^6 is added to Hs1 for rounding off, and it is set to Hs2.

Finally, H is calculated by shifting Hs2 to the 6-bit right. And conversion to H, S, and V from R, G, and B is completed.

4. Experiment/Results

4.1 Conversion accuracy

The accuracy of the developed fixed-point version was verified by comparing the processing result of the floating point version. The title should be written on the second line centered. The used picture is Lenna of a standard picture and the size is 256×256. Moreover, it is a color image whose RGB is 8 bits each. The result was 0.314758 when the mean squared error (MSE) of the result of the fixed-point version and the result of the floating point version was calculated. The result was 56.15 dB when the peak signal-to-noise ratio (PSNR) was calculated using this value. Therefore, it can be said that the accuracy of the developed fixed-point version is equivalent to the floating point version.

4.2 Performance

Designed conversion HW was mounted in SPARTAN-6 XC6SLX16-CSG324 of Xilinx. As a result, the clock period in a sequential version was able to operate in 6.385 ns (a clock frequency is 156.6 MHz), and the clock period in the pipeline version was able to operate in 7.665 ns (a clock frequency is 130.5 MHz). A circuit scale is shown in Table 1.

Table 1. The circuit scale at the time of mounting in SPARTAN-6 XC6SLX16-CSG324.

		Slice Registers	Slice LUTs
Used	Sequential	968	811
	Pipeline	1006	847
Available	Sequential	18224	9112
	Pipeline		
Utilization	Sequential	5%	8%
	Pipeline	5%	9%

4.3 Performance comparison

For the performance comparison, the execution time in a personal computer (PC) and the execution time in the inclusion processor were measured. The used picture is the same as the picture at the time of accuracy verification. The processor of PC is Athlom II X2 B26 of an AMD company, and clock frequency is 3.2 GHz. An inclusion processor is Cortex-M4 of an ARM company carried in STM32F407VGT6 of ST micro company, and clock frequency is 168 MHz. The execution time of PC was 1.14 ms and the execution time of the inclusion processor was 35 ms.

About the conversion HW, the divider required 20 clocks, the multiplier required 1 clock, and the number of clocks required for conversion of 1 pixel of a sequential version became 29 clocks in total. Therefore, the time concerning conversion of a 256×256 pixel picture can be estimated at 12.13 ms. The number of clocks required for conversion of 1 pixel of the pipeline version became 24 clocks. Therefore, the time concerning conversion of a 256x256-pixel picture can be estimated at 0.5 ms.

About the sequential version, the performance was 1/11 to PC which operates 20 times faster. But, to the inclusion processor, 2.9 times as much performance gain has been attained. Moreover, about the pipeline version, the performance was twice to PC. In addition, to the inclusion processor, 70 times as much performance gain has been attained. Therefore, the validity of the conversion HW in small-scale embedded devices was shown.

4.4 Performance electric power ratio (SPR)

Moreover, the performance electric power ratio (SPR) to the conversion hardware of PC was calculated. SPR is calculated by formula (4). And power consumption is calculated by formula (5).

$$SPR = \frac{Processing\ time(HW)}{processing\ time(PC)} \times \frac{Power\ consumption(HW)}{Power\ consumption(PC)} \quad (4)$$

$$P_D = \alpha f C V^2 + \alpha f I_S V \quad (5)$$

Here, α is a switching rate. f is clock frequency. C is a load capacitance. V is power supply voltage. I_s is penetration current (integration value). α is the rate how many all the transistors switch during operation. C is the total gate capacitance of all the transistors. I_s is the total current which flows Drain-Source at the time of switching of a transistor. Since these α, C, and I_s are dependent on a device or operation, it are difficult to acquire. Therefore, although it is thought that it becomes advantageous to the PC side, these are omitted this time. About V, HW is 1.2V, PC is 1.35V, and since it is almost the same, V is also omitted. Therefore, power consumption was estimated by making greatly different clock frequency (it is proportional to electric power) into a central value. In sequential HW, processing time is 12.13 ms and clock frequency is 156.6 MHz. In pipeline HW, processing time is 0.5 ms and clock frequency is 156.6 MHz. In PC, processing time is 1.1 ms and clock frequency is 3200 MHz. These values are assigned to a formula (1) and SPR is calculated.

In sequential HW, the ratio of processing time was 11 and the ratio of power consumption was 0.0489. As a result, SPR was 0.5379. Although 11 times performance of PC is good to HW, the electric power of this is also 20 times, and 1.86 times of SPR are worse than HW. That is, 1.86 times of HW were better than PC about SPR.

In pipeline HW, the ratio of processing time was 0.45 and the ratio of power consumption was 0.0408. As a result, SPR was 0.0184. The performance of PC is 0.45 times to the HW, the electric power of this is 24.5 times, and 53.8 times of SPR are worse than HW. That is, 53.8 times of HW were better than PC about SPR.

Therefore, about both the sequential version and the pipeline version, the result that it was better than PC was obtained.

5. Conclusions

The method of fixed-point-izing at the time of hardware-izing a HSV conversion process and the details of hardware were shown. By verification of accuracy, it checked that there was accuracy equivalent to the floating point version, and the validity in small-scale embedded devices was shown through the quality assessment. From now on, reduction of a circuit scale and verification of operation with the existing equipment will be performed.

References

[1]Tim Kaulmann, Marc Strunkmann, and Ulf Witkowski, "FPGA-based Object Detection in Robot Soccer Application", Proceedings of the 3rd International

Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE 2005), pages 135-140, 2006

[2]Ming Z. Zhang, Li Tao, Ming-Jung Seow, and Vijayan K. Asari, “Design of an Efficient Flexible Architecture for Color Image Enhancement”, Lecture Notes in Computer Science, Volume 4186, Pages323-336, Advances in Computer Systems Architecture, 2006

[3]Junguk Cho, Bridget Benson, Sunsern Cheamanukul, and Ryan Kastner,
”Increased Performance of FPGA-Based Color Classification System”, FCCM ’10 Proceedings of the 2010 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines, Pages29-32, 2010

[4]M.Holzer, B.Knerr, P.Belanovic, and M.Rupp, “Efficient Design Methods for Embedded Communication Systems” EURASIP Journal on Embedded Systems, Volume 2006, Article ID 64913, Pages1-18, Hindawi Publishing Corporation, 2006

[5]Koschan, A. and Abidi, M, “Digital Color Image Processing”, John Wiley & Sons, Pages60–62, 2008