CMOS 이미지 센서용 효과적인 Effect 구현

송형돈, 이동훈, 손승일
한신대학교 정보통신학과

Implementation of Efficient Effect of CMOS Image Sensor

Hyung Don Song, Dong Hun Lee, Seung Il Sonh
Dept. of Information and Communication Hanshin University
e-mail : brothermoney@hs.ac.kr

요 약

본 논문에서는 영상 입력 장치 또는 카메라 이미지 센서로부터 얻은 이미지 데이터에 대하여 Bayer입력 포맷을 우리가 디스플레이 장치로 보는 영상으로 출력하기 위해 전처리 작업을 수행한 후 이미지 이펙트를 수행한다. 본 논문에서는 연산량과 레지스터의 수를 줄이고 칩의 성능을 향상시키기 위해 이미지 이펙트를 RGB에 적용하지 하지 않고 YCbCr을 이용하여 이미지 이펙트를 수행한다. 이를 구현하기위해 원본 이미지 사이즈를 640x480으로 입력 데이터를 사용하고, 소프트웨어로 전처리하여 이미지 결과를 확인한 후, 최적화된 알고리즘를 적용하여 VHDL설계언어를 이용한 하드웨어 설계 후, ModelSim 6.0a를 이용하여 데이터를 검증한다.

1. 서 론

요즘 디지털 카메라나 카메라폰의 사용자는 복합된 기능과 고화질, 소형화, 저비용, 저전력 등을 요구하는 추세이며, 이제 경쟁력 있는 제품을 기업들은 단기간에 생산하여 소비자 욕구를 충족시키고자 한다. CMOS 이미지 센서를 통해 들어온 데이터에 대해 ISP(Image Signal Processing) 처리에서 다양한 기능을 처리하는데 그 중에 감마정정, 컬러인터폴레이션, 컬러정정, 컬러 공간적 변환, 이미지 스케일(확대, 축소), Image Effect(엠보싱, 반전, 세피아)처리 등 다양한 전처리 기능을 수행해야 한다[1][2]. 본 논문에서 이미지는 한 픽셀이 RGB로 이루어져 있는데 이 RGB에 이미지 이펙트를 적용할 경우 많은 연산량과 다수의 레지스터가 필요하므로 처리 시간이 오래 걸라며 비효율적이다. RGB의 데이터 비율은 8:8:8이고 YCbCr의 데이터 비율은 4:2::2의 비율로 압축 하여 전송하므로 YCbCr로 이미지 이펙트를 처리한다. 따라서 RGB 데이터 비트율이 적은 YCbCr의 픽셀값을 ITU-601에서 제공하는 RGB → YCbCr변환식을 이용하여 구하고 수학적으로 계산한 후 소프트웨어 검증을 통해 YCbCr로 이미지 이펙트를 처리함으로써 좀 더 효과적인 처리와 수행속도의 향상을 가져온다. 이펙트 처리 과정에서 RGB 데이터 비트율 보다 YCbCr 변환된 데이터 비트율이 더 작아서 하드웨어 설계 시 더 적은 면적을 차지하므로 설계하기에 적합하고 또한 효과적이다. 두 가지 방법에서 RGB와 YCbCr에 대한 각 픽셀값을 구한 후에 각각의 이미지 효과처리를 프로그램으로 코딩하여 처리한 후 이미지를 확인한 결과 YCbCr을 사용한 이미지 효과가 효율적이면서 연산량이 적어 수행 속도의 향상이 있었다. 소프트웨어 검증을 통해 이를 최적화된 알고리즘으로 VHDL 하드웨어 설계언어를 이용하여 코딩한 후 ModelSim6.0a를 이용하여 파형을 검증하고 원본 이미지와 대상이미지에 대한 이미지를 비교하여 설계에 대한 결론을 내리고자 한다[1][2].

2. 이미지 센서의 전처리(PreProcessing) 개요

2-1. 컬러 공간적 변환

아래 그림 1은 컬러공간적변환 처리 과정에서 사용하는 RGB를 YCbCr로 변환되는 식을 보여주고 있다.

$$Y = 0.299R + 0.587G + 0.114B$$
$$Cb \approx (B-Y)/1.772 + 128$$
$$\approx -0.168736R - 0.331264G + 0.5B + 128$$
$$Cr = (R-Y)/1.402 + 128$$
$$= 0.5R - 0.418688G - 0.081212B + 128$$

ITU-601(formerly CCIR-601)

그림 1. RGB를 YCbCr로 변환

RGB 픽셀을 위 그림 1과 같이 ITU-601의 영상 표준변환식에서 제공해 주는 계산식을 이용하여 YCbCr을 구한다. 여기서 기존 RGB 픽셀에 적용한 픽셀값을 ITU -601의 변환식에서 도출하여 YCbCr값의 각 픽셀을 구하여 본 논문에서 구현하고자 하는 이미지 효과처리에 사용하는 고정 계수값을 구한다[2][3].

2-2. ISP 처리 블록도

영상 입력으로 들어오는 데이터는 감마정정, 컬러 인터폴레이션, 컬러정정, 컬러 공간적 변환, 화이트 밸런스, AE (Auto Exposure), BLC(Black Level Correction) 등과 같은 기능을 기본적인 ISP처리 기능에서 수행하는데 사람의 눈은 완벽하게 수행 할 수 있지만, 카메라 렌즈는 ISP처리를 해야만 사람이 볼 수 있는 이미지가 된다.

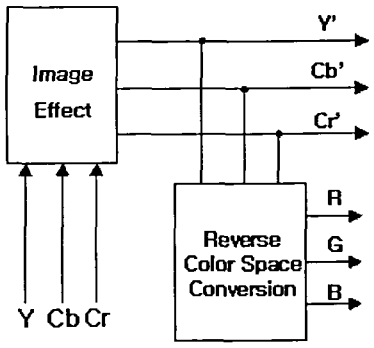

그림 2. ISP전처리 과정 블록도

그림 2는 전처리 과정에서 YCbCr 데이터 입력을 받아서 새로운 Y'Cb'Cr'로 변환되어 이미지 효과 처리 과정을 보여주고 있다. 또한 이미지 효과를 처리한 후 출력포맷은 Back_end 처리단의 요청에 따라 RGb 또는 YCb Cr 포맷으로 출력한다[3][4][5].

2-3. 엠보싱 효과

아래 그림 3은 엠보싱 효과는 YCbCr 중 Y값만을 가지고 적용한다.

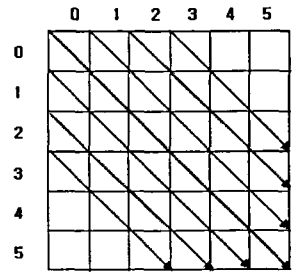

그림 3. 엠보싱 효과 처리

첫 번째 라인의 데이터 픽셀은 그대로 사용하고 두 번째 라인을 처리하기 위해서 첫 라인과 세 번째 라인을 이용하는데 첫 라인 픽셀의 데이터와 현재 라인 데이터와의 대각선 형태로 일치하는 데이터만 처리한다. 나머지 처리도 이와 동일하게 처리한다[1][2].

다음은 각 픽셀 위치별 계산식이다.

단, (*iff* $1<x<5$, $1<y<5$) 조건식일 때

$$Y(x,y) = Y(x-1,y-1) - Y(x+1,y+1)$$

2-3. 이펙트 처리 블록도

제안한 그림 3에서는 엠보싱 효과만 라인 버퍼를 사용하기 때문에 라인 버퍼를 사용하지 않는 효과 처리와 라인 버퍼를 사용하는 효과 처리를 나누어 처리하는 블록도이다. 두 부분으로 나누어 처리를 함으로써 이미지 효과 처리 속도를 향상 시킬 수 있다[2][3].

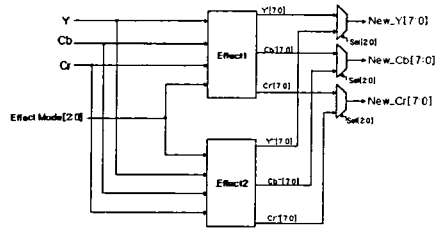

그림 4. 이미지 효과 처리 블록도

3. 이미지 이펙트 블록의 설계

3-1. 이미지 이펙트1 블록도

제안한 블록 그림 5에서는 엠보싱 효과가 아닌 나머지 효과 아쿠아(Cool), 바이올릿(Violet), 세피아(Sepia), 반전(Invert), 미백효과, 그레이 등과 같은 효과 처리를 수행하는 블록도를 나타낸다.

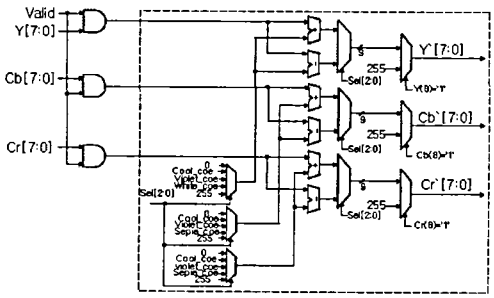

그림 5. 엠보싱을 제외한 처리 블록도

이미지 효과에 따른 특정 픽셀값을 ITU-601의 변환 수식을 이용하여 각 YCbCr값에 고정 픽셀값을 구한다. 효과가 적용된 YCbCr 값을 구하기 위해 변환식에서 수학적 처리를 한 후에 소프트웨어로 검증을 마친 이미지 계수들을 사용한다. 이펙트 모드를 인코딩한 Sel[2:0] 신호에 의해서 이미지 계수들이 선택되어지고 입력 신호 YCbCr과 이미지 계수들은 Sel[2:0]신호에 의해서 덧셈이나 뺄셈을 수행한 뒤 덧셈의 경우 오버플로가 발생할 경우 출력 신호값을 255로 한다. 덧셈기와 뺄셈기, 모드에 따른 연산결과를 나눌 Mux, 오버플로가 발생했을 때 사용할 Mux로 구성한다[4][5].

3-2. 이미지 이펙트2 블록도

제안한 그림 6에서는 엠보싱 효과를 수행하는 처리 블록도이다. 2개의 라인버퍼를 사용하여 이전 데이터 라인과 현재 데이터 라인을 연산하기 위해서 사용하며 첫 라인과 마지막 라인을 제외한 각 라인별 처리를 수행한다. 연산에 필요한 라인버퍼 2개와 뺄셈기와 덧셈기를 있으며 연산값의 오버플로가 발생 하였을 때와 최종 출력단의 Y값을 처리하기위한 데이터 선택 Mux를 두어 처리한다.

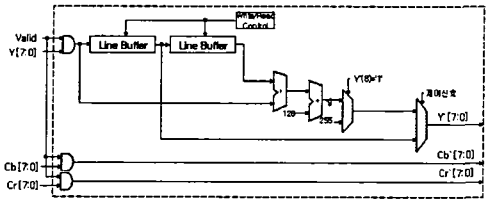

그림 6. 엠보싱 효과 처리 블록도

4. 파형 분석

그림 7은 이미지 이펙트1의 블록을 처리하여 수행하는 파형 결과이다.

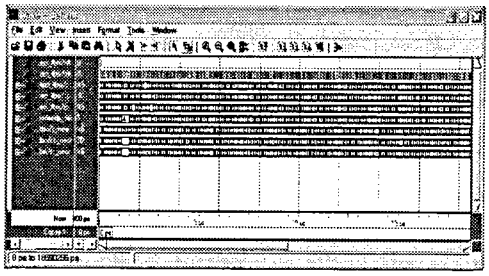

그림 7. 이미지 이펙트1 파형 시뮬레이션

다음 파형 분석에 필요한 이미지 정보는 원본 이미지 사이즈 (640x480)이고, 입력 데이터는 YCbCr 포맷이다. 그림 7은 본 논문에서 제안한 이미지 이펙트1 블록을 수행한 결과이다. 입력 데이터로부터 출력 데이터는 1클럭 이후 연속적으로 수행한 결과를 도출하였다.

그림 8은 본 논문에서 제안한 이미지 이펙트2

블록을 수행한 결과이다. Valid 신호가 '1' 일 때 유효 데이터가 발생하고 처음 데이터 출력은 2 번째 라인이 들어온 후 2 클럭 이후부터 연속적인 처리 결과를 출력한다.

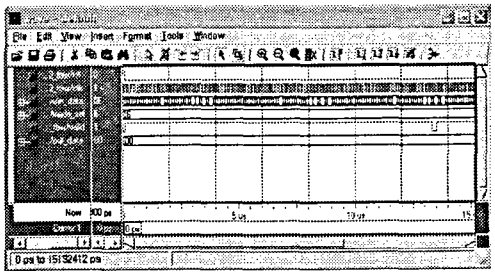

그림 8. 이미지 이펙트2 파형 시뮬레이션

5. 설계 결과

본 논문의 수행 결과를 테스트하기 위해 그림 10는 같은 원본 이미지 그림 9에 대하여 RGB 포맷을 이용하여 이미지 이펙트를 처리한 것과 본 논문에서 제안한 방식으로 YCbCr 포맷을 이용하여 이미지 이펙트를 처리한 결과를 보여주고 있다[3][4].

그림 9. 원본 이미지

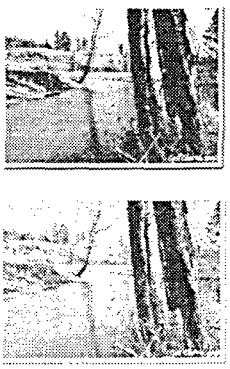

그림 10. 이펙트 처리결과

또한 PSNR식에서 도출한 값이 원본 이미지와 두가지 방법으로 처리하여 비교한 결과 PSNR값의 차이가 거의 없었다. 결국 YCbCr를 이용한 이미지 효과 처리가 적은 데이터 비트율을 가지며 또한 RGB와 비교하여 차이가 거의 없는 효과적인 이미지를 처리 할 수 있었다.
제안한 블록의 각 등가게이트수와 타이밍 결과는 표-1 회로합성 결과표와 같이 나타났다.

표-1 회로 합성 결과

Block	No. of Gates	Timing	Target Device
Image Effect1	1,374	Mininum Period : 4.208ns Max. Frequency : 237.643MHz	XCV1000e-hq240
Image Effect2	67,719	Mininum Period : 10.444ns Max. Frequency : 95.749MHz)	

6. 결론

카메라 이미지 센서용, 영상 입력 장치로부터 얻은 Bayer 데이터를 ISP 전처리 과정을 거쳐 이미지 이펙트 처리에서 RGB보다 적은 연산과 적은 레지스터를 가지고 이미지 이펙트 처리로 하드웨어 설계 효율을 높였고 이미지 결과에서도 좋은 영상 이미지를 도출 할 수 있었다. 결과적으로 카메라 이미지 센서용 ISP(Image Singal Pro cessing)처리 적용 칩에서 제안한 알고리즘이 효과적으로 처리 할 수 있을 것으로 사료 된다[1][2].

7. 참고문헌

[1]. Yun Ho Jung, Jae Seok Kim, Bong Soo Hur and Moon Gi Kang, Department of Electronic Engineering Yonsei University, Seoul, Korea, "Design of Real-Time Image Enhancement Preprocessor for CMOS Image Sensor", IEEE Trans. Consumer Electronics, Vol. 46, No 1, February 2000.

[2]. Bongjun Lee, Jaeseok Kim and Chulhee Lee, Dept. Electric and Computer Engineering,

Yonsei University, 134 Shin chon-Dong, Seodaemoon-Gu, Seoul 120-749, Korea, "High Quality Image Interpolation for Color Filter Arrays" IEEE Trans. 2000.

[3]. Marc J. Loinaz, Member, IEEE, Kanwar Jit Singh, Member, IEEE, Andrew J. Blankby, Student Member, IEEE, David A. Inglis, Kamran Azadct, Member, IEEE and Bryan D. Ackland, Fellow IEEE. "A 200-mW, 3.3V, CMOS Color Camera IC Producing 352x288 24-b Video at 30 Frames/s. IEEE JOURNAL OF SOLID-STATE CIRCUITS, Vol. 33, No. 12, December 1998.

[4]. Randy Crane, Hewlett-Packard Company. "A Simplified

Approach to Image Processing". Prentice Hall PTR

[5]. Rafael C. Gonalez, Richard E. Woods. "Digital Image

Processing". Addison Wesley.