

# CMOS 이미지 센서를 위한 실시간 전처리 프로세서의 설계

정 윤 호(丁 胤 豪), 이 준 환(李 俊 煥), 김 재 석(金 在 錫)

연세대학교 전기·컴퓨터공학과

전화 : (02) 361-4018 / 팩스 : (02) 312-4584

## A Design of the Real-Time Preprocessor for CMOS image sensor

Yunho Jung, Joonhwan Lee, and Jaeseok Kim

Comm. ASIC Design Lab., Dept. of Electrical and Computer Engineering, Yonsei Univ.,

134 Shinchon-Dong, Seodaemun-Gu, Seoul, Korea, 120-749

E-mail : yhjung@asic.yonsei.ac.kr

### Abstract

This paper presents a design of the real-time preprocessor for CMOS image sensor suitable to the digital camera applications. CMOS image sensor offers some advantages in on-chip integration, system power reduction, and low cost. However, it has a lower-quality image than CCDs. We describe an image enhancement algorithm, which includes color interpolation, color correction, gamma correction, sharpening, and automatic exposure control, to compensate for this disadvantage, and present its efficient hardware architecture to implement on the real-time processor. The presented real-time preprocessor was designed using VHDL, and it contains about 19.2K logic gates. We also implement our system on FPGA chips in order to provide the real-time adjustment and it was successfully tested.

### I. 서론

우수한 성능의 영상 취득 소자를 사용하는 것은 고성능 영상 취득 장치 개발에 있어서 필수적이다. 그러나, 이러한 소자는 영상 해상도의 증가에 따라 기하급수적으로 비용이 상승하며, 특히 어느 정도 이상으로는 물리적인 특성을 향상시킬 수 없는 단점이 있다. 이러한 문제점을 디지털 영상 처리 기법을 이용하여 보완할 수 있다면, 비용의 상승을 최대한 억제하면서

고성능의 영상 취득 시스템을 구현할 수 있다. 특히 최근 기존의 CCD(Charge coupled Device) 소자를 대체할 수 있는 영상 취득 소자로 주목받고 있는 CMOS 센서 소자의 경우, CCD에 비해 상대적으로 열세인 성능을 디지털 영상 처리 기법으로 보완함으로써 비용 및 주변 기기와의 통합 가능성과 같은 CMOS 소자의 여러 가지 이점을 효과적으로 활용할 수 있을 것이다 [1].

본 논문에서는 이미지 센서에 의해 획득된 영상의 화질을 디지털 영상 처리 기법을 이용하여하는 전처리 알고리즘을 개발하고 이의 효율적인 실시간 하드웨어 구조를 제안한다. 2장에서 CMOS 이미지 센서를 위해 개발된 전처리 알고리즘이 설명되고, 3장에서는 알고리즘을 구현하기 위한 하드웨어 구조가 제시된다. 4장에서는 실험 결과에 대해 논의한다. 마지막으로 5장에서 결론을 맺는다.

### II. CMOS 이미지 센서를 위한 전처리 알고리즘

그림 1은 CMOS 이미지 센서를 위한 전처리 알고리즘의 기능블록도를 보여준다. 전처리 알고리즘은 크게 5개의 기능블록으로 구성된다. 우선, 이미지 센서로부터의 입력 데이터는 색상 보간(Color Interpolation) 블록을 통해 RGB 픽셀 데이터로 변환된다. 이는 일반적인 이미지 센서가 그림 2와 같은 Bayer 포맷의 칼라 필터 어레이를 갖고 있기 때문이다.

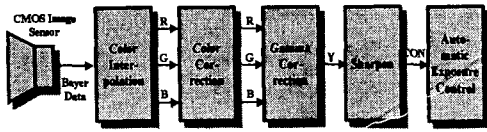


그림 1. 전처리 알고리즘의 기능블록도

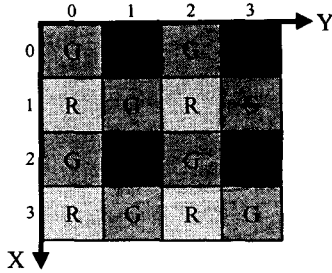


그림 2. RGB 칼라 필터 어레이

색상 보간은 주변 픽셀값들의 평균을 통해 R, G, B 각각의 값을 구하는 Nearest Neighbor Averaging 방식으로 구현되었다[2].

색상 보간 과정을 거쳐 생성된 RGB 픽셀 데이터는 단지 주변값들의 평균을 통해 얻어졌으므로 색상 보정 과정을 통해 실제 색을 재현해내야 한다. 색상 보정 과정을 구현하는 방법은 크게 두가지로 나뉘어 질 수 있다. 첫째는 색상 좌표계에서 실제 색상과 현재 나타난 색상간의 오차를 보정해주는 방법이고 둘째는 현재 색상을 특정 목적에 의해 다른 색상으로 임의로 지정해주는 방법이다. 즉 현재 색상을  $c(R, G, B)$ 으로 표시할 때, 이 값을 새로운 색상인  $c'(R', G', B')$ 로 변환하는 방법이며 (1)과 같은 식으로 표현할 수 있다[3].

$$c' = f(A, c, k) = A \cdot c + k \quad (1)$$

위 식에 의해 원래의 색상 행렬  $c$ 는 변환 행렬  $A$ 에 의해 새로운 색상 행렬  $c'$ 로 변환된다. 따라서 식(2)와 같은 행렬 연산을 통해 색상 보정 알고리즘을 구현할 수 있다. 행렬식의 각 원소들은 실제 색상을 재현해 낼 수 있도록 여러 차례의 모의 실험을 통해 결정되었다.

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2)$$

CRT(Cathod Ray Tube)가 입력 영상의 전압에 비례하는 밝기(Intensity)를 만들어내지 못하기 때문에 모니터상의 영상과 실제 영상사이에는 차이가 존재하게 된다. 오히려 CRT에 의한 밝기는 입력 전압의 감

마( $\gamma$ )승에 비례하기 때문에 식(3)에서와 같이 입력 신호에  $1/\gamma$  승을 취함으로써 감마 보정을 수행할 수 있다[4].

$$R_{gc} = 255 \times \left( \frac{R}{255} \right)^{\frac{1}{\gamma}}$$

$$G_{gc} = 255 \times \left( \frac{G}{255} \right)^{\frac{1}{\gamma}} \quad (3)$$

$$B_{gc} = 255 \times \left( \frac{B}{255} \right)^{\frac{1}{\gamma}}$$

이미지 센서에 의해서 촬영된 영상의 Dynamic range는 CRT Display luminance(약 20dB)보다 훨씬 높다. 게다가 영상 밝기의 최대, 최소값도 시간에 따라 변한다. Global automatic gain/bias control(AGC)은 영상의 양극단이나 전반적인 영상의 분산에 의해서 달라지는 밝기를 선형적으로 scaling하는 것으로 구현될 수 있다. 그러나 영상의 Dynamic range가 큰 경우 단지 global automatic gain/bias control만으로 화질이 우수한 영상을 얻기는 어렵다. Dynamic range가 큰 영상을 한정된 Dynamic range를 가진 디스플레이를 통해 보면 낮은 contrast local detail을 가진 부분은 알아볼 수 없게 되기 때문이다. 따라서 이런 현상을 해결하기 위해 전반적인 영상의 Dynamic range를 디스플레이에 맞게 압축을 하는 반면 local contrast는 향상시켜 주는 local area contrast enhancement(Sharpening)가 필요하다.

AEC(Automatic Exposure Control)란 주변 환경에 따라 노출 시간을 자동으로 변화시켜 들어오는 빛의 양을 조절하는 것을 말한다. 일정하게 빛의 양이 조절되면 주변 밝기와는 관계없이 일정한 밝기의 영상을 얻을 수 있다. CMOS 이미지 센서가 어두운 곳에 있다면 노출 시간을 늘려주어야 밝은 화면을 얻을 수 있고, 밝은 곳에 있다면 노출 시간을 줄여주어야 적당한 밝기의 화면을 얻을 수 있다. 이는 노출 시간과 들어오는 빛의 양이 비례관계에 있기 때문이다. 일반적인 CMOS 이미지 센서는 내부에 노출 시간 제어 레지스터를 가지고 있기 때문에 적절한 임계값을 정한 후 입력된 영상의 중간값(Median)과 비교를 해서 중간값이 임계값보다 크다면 노출 시간을 줄여주고 중간값이 임계값보다 작다면 노출 시간을 늘려주는 방식으로 AEC 알고리즘을 구현할 수 있다.

### III. 전처리 프로세서의 하드웨어 구조 설계

그림 3은 제안된 실시간 전처리 프로세서의 전체 블록도를 보여준다.

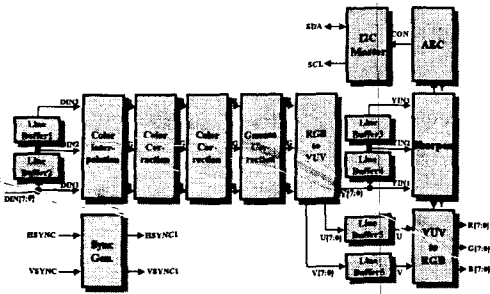


그림 3. 구현된 전처리 프로세서의 전체 블록도

CMOS 이미지 센서로부터의 8Bit 입력 데이터는 RGB 픽셀 데이터를 만들기 위해 색상 보간 모듈을 거친다. II장에서 언급된 것처럼 Nearest Neighbor Average 방법을 구현하기 위해서는 입력 영상 신호의 3×3 윈도우 픽셀값들이 필요하게 된다. 따라서 색상 보간 과정을 위한 모듈은 픽셀값들의 저장을 위한 Line Buffer, 평균값을 구하는 Data Generator, 평균값의 선택을 위한 Mean Selector로 구성되어 있다. 이의 블록도는 그림 4와 같다.

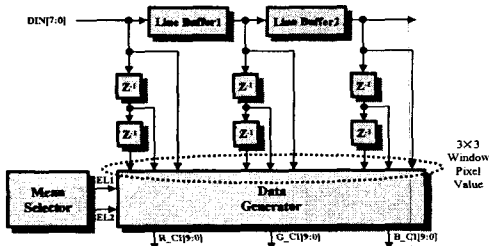


그림 4. 색상 보간 모듈의 블록도

색상 보정 모듈은 식(2)와 같은 행렬 연산을 통해 구현될 수 있다. 하드웨어 설계를 위해 식(2)를 정리하면 식(4)와 같다.

$$\begin{aligned} R' &= b_{11}R + b_{12}G + b_{13}B \\ G' &= b_{21}R + b_{22}G + b_{23}B \\ B' &= b_{31}R + b_{32}G + b_{33}B \end{aligned} \quad (4)$$

식 (4)에서 행렬의 원소에 해당하는  $b_{ij}(i=1-3, j=1-3)$ 는 모의 실험에 의해 결정되었고 이 연산은 쉬프터와 덧셈기에 의해 구현될 수 있다.  $R'$ 을 구하는 블록을 그림 5에서 볼 수 있다.

감마 보정을 수행하는 과정인 식(3)은 모의 실험 결과 결정된 감마값이 1.6이었으므로 식(5)와 같이 변형될 수 있다.

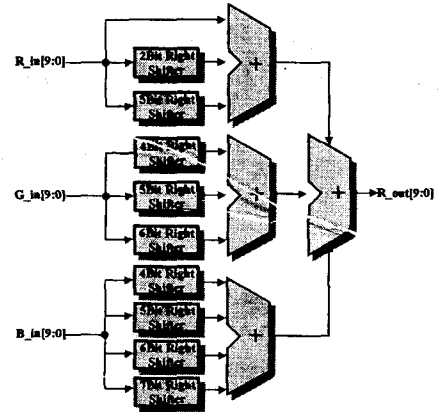


그림 5. 색상 보정 모듈의 블록도

$$\begin{aligned} R_{gc} &= 255 \times \left(\frac{R}{255}\right)^{\frac{1}{1.6}} \\ &= 255 \times \left(\frac{R}{255}\right)^{\frac{1}{1.6}} \\ &= 255^{\frac{3}{8}} \times (R)^{\frac{5}{8}} \\ &\cong 8 \times (R)^{\frac{5}{8}} \end{aligned} \quad (5)$$

따라서 감마 보정 과정은 입력 신호의 5/8승을 구하는 ROM table과 이의 결과를 8배하는 레프트 쉬프터로 구현할 수 있다.

Sharpening 알고리즘의 구현을 위해 우선 주어진 3×3 윈도우 내의 각 픽셀값들에 대한 평균과 분산을 구하는 블록이 필요하다. 윈도우 내 픽셀값을 얻기 위해 색상 보간 모듈에서와 같이 Line Buffer를 사용하였으며 평균과 분산을 구하는 블록, 구해진 평균과 분산의 비에 따라 Gain을 구하는 블록, 그리고 결과 데이터를 구하는 블록으로 구성되어 있다. 이의 블록도는 그림 6과 같다.

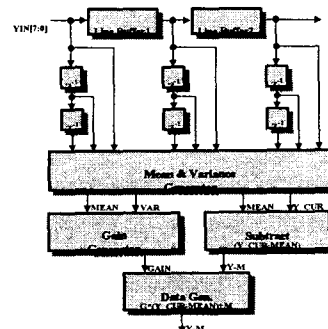


그림 6. Sharpening 모듈의 블록도

AEC 알고리즘은 입력 영상의 Median 값을 프레임 단위로 측정하여 이 값이 임계값(128)보다 크면 I2C Bus Master를 통해 이미지 센서내의 Integration Time Control Register의 값을 낮추어 주고 임계값보다 작은 경우 Register의 값을 크게 하는 것이다. 알고리즘의 구현은 입력 영상의 Median값을 구하는 대신 Counter를 통해 입력 신호(픽셀값)가 165보다 큰 경우를 세고 또한 85보다 작은 경우를 세어 큰 경우의 수가 전체 픽셀수의 1/2이상이면 Control Register의 값을 낮추어 주고, 반대로 작은 경우의 수가 전체 픽셀수의 1/2이상이면 Register의 값을 크게 하도록 하여 이루어 졌다. 따라서 기본적인 하드웨어의 구조는 165보다 큰 경우를 세는 Counter165 블록, 85보다 작은 경우를 세는 Counter85블록, 전체 픽셀수와 비교하는 비교기 블록 그리고 I2C I/F 블록으로 구성되어 있다. 이의 블록도는 그림 7과 같다.

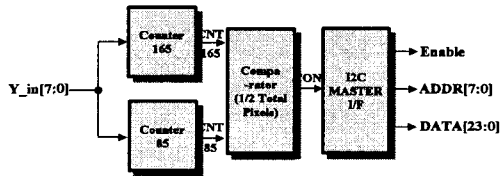


그림 7. AEC 모듈의 블록도

#### IV. 실험 결과

그림 8은 실험에 사용된 입력영상이다. 이를 통해 CMOS 이미지 센서는 그림 2와 같이 Bayer 포맷 칼라 필터 어레이를 갖고 있음을 확인할 수 있다.

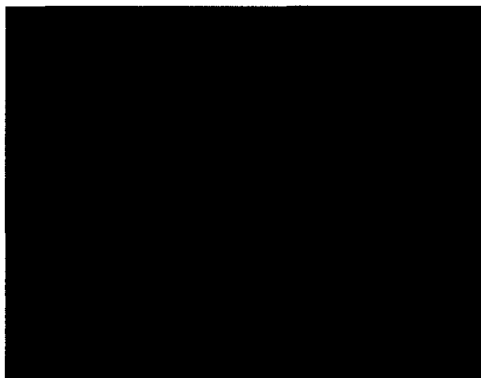


그림 8. 검증에 사용된 입력 영상

그림 9는 개발된 전처리 알고리즘인 색상 보간, 색상 보정, 감마 보정, Sharpening, AEC를 거친 결과 영상이다. 입력 영상인 그림 9에 비해 훨씬 더 실제 영상에 가깝게 개선되었음을 확인할 수 있다.

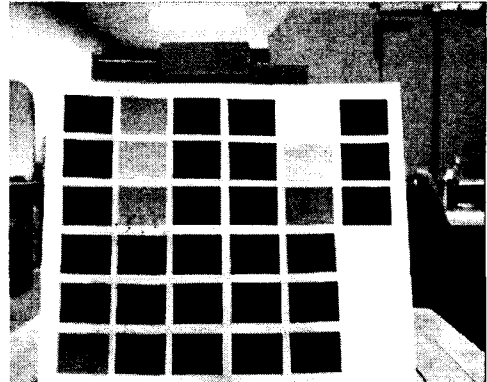


그림 9. 설계된 실시간 전처리 프로세서에 의한 최종 결과 영상

#### V. 결론

본 논문에서 우리는 CMOS 이미지 센서를 위한 실시간 전처리 프로세서의 하드웨어 구조를 제시했다. CMOS 이미지 센서에 의한 영상의 화질 개선을 위해 적합한 전처리 알고리즘이 개발되었고 이의 효율적인 하드웨어 구조가 제안되었다. 논문에서 제시된 전처리 프로세서에 의한 최종 결과 영상이 CCD 소자 수준의 성능을 보임을 확인할 수 있었다. 제안된 하드웨어는 VHDL을 이용하여 설계되었고, 합성 결과, 대략 19K의 로직 게이트를 얻었다. 실험 결과를 얻기 위해 그리고 실시간 모드에서의 동작을 확인하기 위해 설계된 프로세서는 FPGA 칩으로 구현되었고 성공적으로 테스트되었다.

#### 참고문헌

- [1] Eric R. Fossum, "CMOS Image Sensors: Electronic Camera-On-A-Chip", IEEE Trans. On Electron Devices, Vol. 44, No. 10, October 1997, pp. 1689-1698.
- [2] Tadashi Sakamoto, Chikako Nakanishi and Tomohiro Hase, "Software Pixel Interpolation for Digital Still Cameras Suitable for a 32-Bit MCU", IEEE Trans. On Consumer Electronics, Vol. 44, No. 4, November 1998, pp. 1342-1352.
- [3] Jen-Chuan Wang, Der-Song Su, Den-Jen Hwung, and Ji-Chien Lee, "A Single Chip CCD Signal Processor for Digital Still Cameras", IEEE Trans. On Consumer Electronics, Vol. 40, No. 3, August 1994, pp. 476-483.
- [4] <http://www.vtisan.com/~rwb/gamma.html>