쉐이딩 보정 알고리즘과 CMOS 이미지 센싱 시스템 설계

김영빈* · 류광렬*

*목원대학교

Shading Correction Algorithm and CMOS Image Sensing System Design

Young Bin Kim, Conan K. R. Ryu*

*Mokwon University

Email : conan@mwu.ac.kr

요 약

본 논문은 CMOS 센서의 스캔 이미지 향상 위한 기법으로 흑백의 명암차를 이용한 이미지 보상 알고리즘과 시스템 설계에 관한 연구이다. 제안한 기법은 CMOS 센서의 각 화소에 대해 명암차를 구하고, 제한된 메모리 조건 하에서 쉐이딩 알고리즘을 적용한다. 제안 기법의 성능은 하드웨어를 설계하여 평가 한다. 실험 결과는 테스트 이미지의 에지 부분에서 채널별 밝기 차이가 개선되었고, 균등한 품질의 이미지 센싱이 가능하였다.

ABSTRACT

The image correction algorithm and system design for CMOS sensor to enhance the image resolution is presented in this paper. The proposed algorithm finds out the image cell from the sensor and process them by the limited memory configuration. The evaluation of the method is done by the designed hardware system. The experimental results are capable of improving contrast per channel and of sensing equalized image quality on an edge of image.

키워드

Shading Correction, Image Enhancement, CMOS Sensor

Ⅰ. 서 론

디지털 카메라, CCTV와 같이 이미지 센서를 이용한 동영상처리 시스템은 사회생활환경에서 광범위하게 활용되고 있다. 이때 획득한 이미지는 주위 환경에 민감하게 영향을 받는다. 스캐너를 사용하여 사진 또는 문서 등을 캡쳐하는 경우도 유사하다. 동일한 사진에 대해 스캔 이미지가 원본과 다른 경우가 있는데, 제조사가 다른 스캐너를 사용한 경우는 그 차이가 더 크게 나타난다. 이러한 차이는 장비의 특성에 의존적이다[2,10] 모든 이미징 시스템은 쉐이딩 현상을 보인다. 쉐이딩의 소스는 이미지를 획득할 때 주변 조명 및 환경에 따라 노출, 초점과 같은 카메라 외부 뿐만 아니라 픽셀과 픽셀 사이의 게인과 오프셋의 변화와 같이 카메라 자체에 있기도 하다. 이러한 다양한 영향이 이미지의 화질에 영향을 주게 된다.[3~5, 8]

이미징 시스템의 쉐이딩 현상은 스캔 이미지의 중앙에서 밝은 반면에 이미지의 에지 부분에 가까울수록 밝기가 감소하는 현상을 흔히 볼 수 있다. 또 다른 경우, 이미지의 오른쪽 부분은 밝고 왼쪽 부분은 어둡게 나타나는 경우도 있다. 이러한 문제점에 대해 기존의 기법은 자주 발생하는 에지 쉐이딩을 위한 문제 해결에 집중을 하고 있다. 또한 기존의 쉐이딩 보정 기법은 회기종류(retrospective type)을 사용하기 때문에 이미 생성된 이미지에만 적용이 가능한 문제점이 있다.[6,7,9] 스캔이미지의 해상도 및 사이즈가 커짐에 따라 이미지 프로세싱에 소요되는 데이터 연산량도 함께 증가한다. 반면에 시스템의 프로세서 성능이 향상 된다 해도 시스템 전체에 대한 데이터 처리의 부하에 영향을 끼친다.[1] 이러한 문제점을 개선하기 위하여 CMOS 이미지 프로세싱이 가능한 쉐이딩 Luck-Up Table(LUT) 알고리즘을 제안하고 FPGA에서 구현하여 성능을 평가하도록 한다.

Ⅱ. 쉐이딩 보정 알고리즘

1.쉐이딩 보정

쉐이딩 보정은 CMOS 센서를 입력 장치로 사용하여 획득한 이미지를 보정하는 과정이다. 센서를 사용하는 카메라, 스캐너 시스템은 센서의 감도와 조도의 상태에 따라 센싱 이미지의 히스토그램 편차가 다르게 나타나기 때문에 균일한 이미지 획득을 위하여 센서 픽셀의 감도와 조도를 쉐이딩 과정으로 보정한다.

2. 보정 알고리즘

본 논문에서는 쉐이딩 보정에 이미지의 음영을 이용한 기법을 사용한다. CMOS 센서의 각 픽셀은 위치에 따라서 조명의 밝기가 다르고 센서의 감도 또한 다르다. 따라서 각 채널별 LUT를 생성하여 픽셀 위치 {m,n}에 대한 보정값으로 사용한다. 그림1은 LUT 생성과 이미지 보정과정을 보이고 있다.

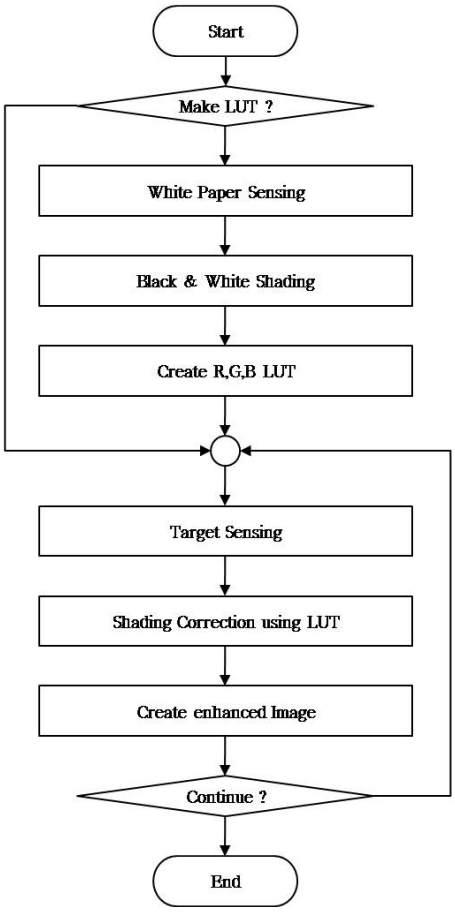


그림 1. 보정 알고리즘

Fig.1 Shading correction algorithm

식(1)은 히스토그램 보정에 사용하는 수식이다.

$$Y[m,n]=\frac{X[m,n]-Black[m,n]}{White[m,n]-Black[m,n]}\times 256 \quad (1)$$

여기에서, 픽셀 m,n 위치의 센서 출력은 $X[m,n]$이고 최소 밝기에서 출력값은 $Black[m,n]$이다. $W[m,n]$은 가장 밝은 조도에서의 출력값이다. $Y[m,n]$은 쉐이딩 보정 결과값으로 256단계 범위 내에 위치한다.

센서의 해상도가 높아지면 처리해야 하는 데이터양도 함께 증가한다. 많은 데이터를 처리하기 위해서는 고가의 고속 프로세서를 사용하거나 신호처리 전용의 프로세서를 사용해야 하는 한계가 있다. 이러한 문제를 해결하기 위해 고속 이미지 데이터 처리를 위한 FPGA 사용한다. FPGA에서는 쉐이딩 연산을 LUT로 만들어 반복적인 이미지 연산에 대한 고속의 처리가 가능하다.

표 1. 픽셀 98~102의 쉐이딩 값

Table. 1 Shading value of 98~102 Pixels.

픽셀 번호	화이트쉐이딩	블랙쉐이딩
:	:	:
98	206	114
99	207	114
100	207	108
101	207	114
102	209	115
:	:	:

표1은 센서 픽셀 98~102 사이의 화이트 쉐이딩과 블랙쉐이딩 데이터 값을 보이고 있다. 표와 같이 각 센서에 대한 쉐이딩 값을 먼저 획득하여 메모리에 저장한다. 앞서 설명한 식(1)을 적용하여 LUT 값 연산 과정을 보인 것이 표2이다.

표 2. A/D 변환값에 대한 LUT값

Table. 2 LUT Values for A/D Result.

	A/D 변환값	LUT 값
Black	0	0
	:	:
↑	108	0
	109	3
	110	5
↓	:	:
White	255	255

픽셀 108에서 LUT 값은 $\frac{108-108}{207-108}\times 256=0$가 된다. 픽셀 109에서의 LUT값은 $\frac{109-108}{207-108}\times 256=3$이 된다. 나눗셈과 같이 여러 개의 인스트럭션을 필요로 하는 연산은 LUT을

만들어 사용함으로서 이미지 캡쳐 과정에서 연산양을 줄일 수 있다.

Ⅲ. CMOS 이미지 센싱 시스템

쉐이딩 보정을 위한 시스템은 그림 2의 설계시스템 블록과 같이 센서 출력에 관련된 CMOS블록, 쉐이딩보상 알고리즘 처리를 위한 쉐이딩 블록, 그리고 LUT 생성 및 시스템 컨트롤을 위한 MPU 블록으로 구성한다.

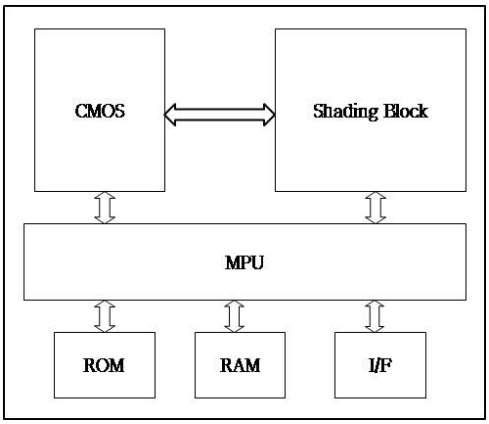


그림 2. 설계 시스템의 블록도
Fig. 2 Realization System Block Diagram.

3.1.CMOS 블록

CMOS블록은 센싱 대상에 대한 아날로그 데이터를 디지털 데이터로 변환하여 출력한다. 출력 데이터는 R,G,B 세가지 컬러에 대한 값이고, 비압축 데이터 포맷을 사용한다.

3.2.쉐이딩 블록

쉐이딩 블록은 센서 각각의 픽셀에 대하여 최대 밝기값과 최소 밝기값을 사용하여 센싱 데이터를 평활화 한다. 쉐이딩 보정 기법은 센서 위치에 대한 감도 또는 조도 차이가 발생하더라도 균등한 히스토그램 값을 갖도록 향상 처리 한다. 히스토그램 향상과정은 최대 밝기값을 사용한 화이트쉐이딩과 최소 밝기값을 사용한 블랙쉐이딩 값을 이용한다. 식(1)에서 보였듯이 뺄셈과 나눗셈 연산을 모든 센서 위치에 대해 반복하고 소프트웨어로 처리하게 되면 많은 시스템 부하로 작용한다. 따라서 FPGA에 알고리즘은 시스템 부팅시 쉐이딩 메모리에 초기화 한다. 초기화 데이터는 각 컬러데이터를 채널별로 분리하고 LUT을 만든다. LUT 데이터는 쉐이딩 블록에 있는 휘발성 메모리에 저장하고 이미지 보정에 사용하도록 한다. 이미지 센싱 동작에서는 센서에서 획득한 데이터를 쉐이딩블록의 입력으로 하고 보정된 출력 데이터만을 사용하여 보정 이미지를 만든다.

쉐이딩 블록은 FPGA와 LUT 데이터를 저장하기 위한 RAM으로 구성된다. FPGA는 변환 데이터를 MPU와 동기화된 시퀀스 과정을 거쳐 데이터를 MPU로 전송하도록 하며, 전송이 완료되면 하나의 프레임 이미지가 완성된다.

3.3.MPU 블록

MPU 블록은 LUT 생성과 시스템 제어에 관련한 동작을 수행한다. LUT 생성은 앞에서 설명한 알고리즘을 기반으로 LUT 데이터 계산에 관련한 연산 동작이다. 시스템 제어 동작은 시스템 초기화 과정, 스캐닝 프로세스에 필요한 처리 동작이다. 시스템 초기화 과정은 시스템에 전원이 인가되면 쉐이딩 블록의 FPGA에 VHDL 코드를 로딩하고, 생성한 LUT 데이터를 업로드 한다.

Ⅳ. 실험 및 고찰

제안 알고리즘은 VHDL을 사용하여 구현하였다. 이미지 센서는 3M CMOS 센서를 사용하였고, 흰색의 백상지를 센싱 입력으로 사용하였다. 조명은 자체 광원을 사용하였다. 표3은 쉐이딩 보정알고리즘 구현을 위한 구현 시스템의 규격이다. FPGA의 게이트 수는 14,579이고 내부의 블록램은 576이다. 기준 클럭은 66.6Mhz을 사용하였다.

표 3. FPGA 규격
Table 3. FPGA Specification.

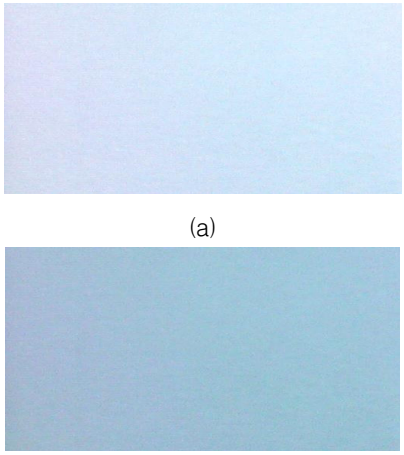
항 목	규 격
FPGA Gate Count	14,579
Block Ram	576
Clock	66.6Mhz
Memory(External)	32Mb*16

그림3은 쉐이딩 보정 알고리즘을 적용하여 얻은 이미지 이다. (a)는 이미지 센싱 후의 원본 이미지 이고 (b)는 쉐이딩 보정 후의 이미지 이다. 보정 후의 이미지가 원본 이미지보다 전체적인 밝기가 낮아지는 것이 보인다. 이것은 LUT를 적용하여 히스토그램 평준화가 이루어졌기 때문이다. 쉐이딩 보정을 한 결과 이미지는 원본 이미지에 비하여 밝기의 차이가 채널별 최대 32%까지 줄어들었다.

Ⅴ. 결 론

본 논문에서는 CMOS 센서를 이용하여 획득한 이미지 향상을 위한 쉐이딩 보정 알고리즘을 제안하였고, 이미지 센싱 시스템을 설계하였다. 쉐

이딩 보정 알고리즘은 센서의 화소에 대해 화이트 및 블랙 쉐이딩 값을 적용한 보정값을 LUT화 하여 사용하였다. 보정 LUT 값은 이미지 캡쳐 동작 전에 캡쳐 환경에 적합한 보정값을 획득하여 사용하였다. LUT 데이터는 쉐이딩 블록에 있는 메모리에 저장한다. 센서 출력 데이터는 각 픽셀 위치에 대응되는 LUT 보정값으로 변환하고 세 개의 컬러 채널을 모아 보정한 이미지를 완성하였다. 이러한 이미지 처리 시스템은 쉐이딩 보정 알고리즘의 복잡한 연산을FPGA와 LUT 메모리로 구성된 하드웨어 블록에서 처리 하였다. 제안 알고리즘을 이미지 센싱 시스템을 구현하여 실험한 결과 채널별 밝기 차이가 개선되었고 균등한 품질의 이미지 센싱이 가능하였다. 또한 복잡한 신호처리 블록이 없는 범용 프로세서를 사용하여 간단한 시스템 구현이 가능할 것으로 기대한다.



(a)

(b)

그림 3. 쉐이딩 보정 결과
(a) 원본 이미지, (b) 쉐이딩 보정 이미지

Fig. 3 Shading correction Results
(a) Original image, (b)Shading Correction Image

참고문헌

[1] 김영빈, 류광렬, “LUT 쉐이딩 보정 알고리듬을 이용한 스캐닝 이미지 향상 FPGA 설계 구현,” 한국정보통신학회논문지, 16권 8호, pp.1759-1764, 2012.

[2] Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, Prentice-Hall, 2002.

[3] 조맹섭, 디지털 컬러 프로세싱, 국제, 2006.

[4] Jooyoung Ha, Sungmok Lee, WonWoo Jang, Hoongee Yang, Bongsoon Kang, “VLSI Implementation of Adaptive Shading Correction System Supporting Multi-Resolution for Mobile Camera,” KICS, vol.31, no.12C, pp. 1201 - 1207, Dec. 2006.

[5] Hyun-Sang Park, “Low-Complexity Lens-Shading Correction Algorithm based on Piecewise Linear Model,” KI-IT, vol.10, no.2, pp. 183 - 189, Dec. 2012.

[6] T. Young, “Shading correction: compensation for illumination and sensor inhomogeneities,” in Current Protocols in Cytometry, J. P. Robinson, et al., Ed., vol. 1, pp. 2.11.1 - 2.11.12, John Wiley & Sons, New York, NY, USA, 2000.

[7] Do-Hyeon Kim, Ho-Young Jung, Eui-Young Cha, “Contrast Enhancement Technique by Intensity Surface Stretching,” KICS, vol.11, no.12, pp. 2398 - 2405 , Dec. 2007.

[8] T E Marchant, C J Moore, C G Rowbottom, R I Mackay, P C Williams, “Shading Correction Algorithm for Improvement of Cone Beam CT Images in Radiotherapy,” Marchant et al. Physics in Medicine and Biology 53, pp.5719-5733, 2008.

[9] 류광렬, 김영빈, “스테레오 라인 CCD를 이용한 이동객체감지 및 경로추적 시스템 구현” 한국정보통신학회논문지, 12권, 11호, pp.2050-2056, 2008.

[10] 김도현, 정호영, 차의영, “명도 표면 스트레칭에 의한 화질 개선 기법” 한국해양정보통신학회논문지, 11권, 12호, pp.2398-2405, 2007.