

---

# LUT 웨이딩 보정 알고리즘을 이용한 스캐닝 이미지 향상 FPGA 설계 구현

김영빈\* · 류광렬\*\*

## FPGA Design and Realization for Scanning Image Enhancement using LUT Shading Correction Algorithm

Young Bin Kim\* · Conan K. R. Ryu\*\*

### 요 약

본 논문은 CCD 센서의 스캔 이미지 향상을 위해 웨이딩 보정 알고리즘을 이용한 FPGA 설계 및 구현에 관한 연구이다. 웨이딩 보정 기법은 룩업테이블(LUT)을 적용한다. 향상된 이미지 스캔을 위하여 CCD 센서의 모든 픽셀에 대한 히스토그램 최대값 및 최소값을 구하고, 균일한 히스토그램 값을 갖도록 오프셋 데이터의 웨이딩 보정 LUT를 생성한다. 스캔 과정에서 센서의 출력은 보정된 LUT값으로 변환되고 LUT 변환 처리는 FPGA로 구현하여 실시간 처리가 가능하도록 한다. 구현 시스템을 사용하여 실험한 결과 2.4ms 이하의 스캔 타이밍을 충족 할 수 있다. 제안 시스템은 낮은 성능의 프로세서 기반에서 저비용 및 실시간 향상된 이미지 스캔이 가능하다.

### ABSTRACT

This paper describes FPGA design and realization using the shading correction algorithm for a CCD scan image enhancement. The shading algorithm is used by LUT (Look-up Table). The image enhancement results from that the histogram minimum value and maximum with respect to all pixels of the CCD image should be extracted, and the shading LUT is constructed to keep constant histogram with offset data. The output of sensor be converted to corrected LUT image in preprocessing, and the converting system is realized by FPGA to be enabled to operate in real time. The result of the experimentation for the proposed system is showed to take the scanning time 2.4ms below. The system is presented to be based on a low speed processor system to scan enhanced images in real time and be guaranteed to be low cost.

### 키워드

웨이딩 보정 알고리즘, 룩업 테이블, 스캐닝 이미지, FPGA, CCD

### Key word

Shading Correction Algorithm, LUT, Scanning Image, FPGA, CCD

---

\* 정회원 : 목원대학교

\*\* 중신회원 : 목원대학교 (교신저자, conan@mwu.ac.kr)

접수일자 : 2012. 06. 28

심사완료일자 : 2012. 07. 17

## I. 서 론

스캐너는 사진, 그림 기타 문서 등을 스캔하여 디지털 이미지 데이터로 변환한다. 스캔 이미지는 원본과 비교해 보면 차이가 있음을 흔히 경험한다. 제조사가 다른 스캐너를 사용한 경우는 그 차이가 더 크게 나타난다. 이러한 차이는 장비의 특성에 의존적이다.[1,2] 모든 이미징 시스템은 웨이딩 효과를 보인다. 웨이딩의 소스는 이미지를 획득할 때 주변 조명 및 환경에 따라 노출, 초점과 같은 카메라 외부환경 뿐만 아니라 픽셀과 픽셀 사이의 개인과 오프셋의 변화와 같이 카메라 자체에도 있다. 이러한 다양한 영향이 이미지의 화질에 영향을 주게 된다.[3~5]

스캔 이미지의 중앙은 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 웨이딩을 위한 문제 해결에 집중을 하고 있다. 또한 제한한 웨이딩 보정 기법은 회기종류(retrospective type)을 사용하기 때문에 이미 생성된 이미지에만 적용이 가능한 문제점이 있다.[6,7] 스캔이미지의 해상도 및 사이즈가 커짐에 따라 이미지 프로세싱에 소요되는 데이터 연산량도 함께 증가한다. 반면에 시스템의 프로세서 성능이 향상 된다 하더라도 시스템 전체에 대한 데이터 처리의 부하는 작지 않은 상태이다. 이러한 문제점을 개선하기 위하여 간단하고 저렴하며 실시간 이미지 프로세싱이 가능한 웨이딩 보정 알고리즘에 대해 LUT를 적용 제안하고 FPGA로 설계 구현하여 성능을 평가한다.

## II. 웨이딩 보정 알고리즘

웨이딩 보정은 CCD 센서를 입력 장치로 사용하는 스캐너 시스템에서 균일한 이미지를 획득하기 위해 사용하는 기법이다. 센서의 감도와 조도의 상태에 따라 스캔 이미지의 히스토그램 편차가 다르게 나타나기 때문에 대부분의 스캐너 시스템은 균일한 히스토그램을 갖는 이미지 획득을 위하여 센서 픽셀의 감도와 조도를 고려한 보정을 필요로 한다. 식(1)은 히스토그램 보정에 사용하는 수식이다.

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

여기에서, 픽셀  $m$  위치의 센서 출력은  $X[m]$ 이고 최소 밝기에서 출력값은  $Black[m]$ 이다.  $W[m]$ 은 가장 밝은 조도에서의 출력값이다.  $Y[m]$ 은 웨이딩 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 웨이딩 연산을 LUT로 만들어 반복적인 이미지 연산에 대한 고속의 처리가 가능하다.

표 1. 픽셀 98~102의 웨이딩 값  
Table. 1 Shading values of 98~102 Pixels

픽셀 번호	화이트웨이딩	블랙웨이딩
:	:	:
98	206	114
99	207	114
100	207	108
101	207	114
102	209	115
:	:	:

표1은 센서 픽셀 98~102 사이의 화이트 웨이딩과 블랙 웨이딩 데이터 값을 보이고 있다. 표1과 같이 각 센서에 대한 웨이딩 값을 먼저 획득하여 메모리에 저장한다. 앞서 설명한 식(1)을 적용하여 룩업테이블 값 연산 과정을 보인 것이 표2 이다. 픽셀 108에서 룩업테이블 값은  $\frac{108 - 108}{207 - 108} \times 256 = 0$ 가 된다. 픽셀 109에서의 룩업테이블 값은  $\frac{109 - 108}{207 - 108} \times 256 = 3$ 가 된다. 나눗셈과 같이 여러 개의 인스트럭션을 필요로 하는 연산은 룩업테이블 만들어 사용함으로써 이미지 캡처 과정에서 연산량을 줄인다.

표 2. A/D 변환값에 대한 LUT값  
Table. 2 LUT Values for A/D Results

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

### III. 시스템 설계 및 구현

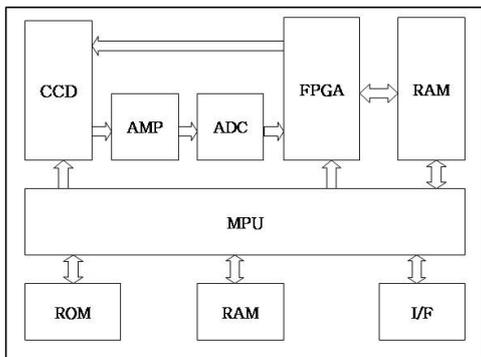


그림 1. 설계 시스템의 블록도  
Fig. 1 Realization System Block Diagram.

웨이딩 보정을 위한 시스템 설계는 센서 출력에 관련된 CCD블록, 웨이딩보정 알고리즘 처리를 위한 웨이딩 블록, 그리고 LUT 생성 및 시스템 컨트롤을 위한 CPU 블록으로 나눈다. 그림1은 구현 시스템의 블록도이다.

#### 3.1. CCD 블록

CCD블록은 센서 데이터를 A/D 변환하는 처리를 하며, 라인 CCD와 AMP, ADC로 구성된다. 각 센서 채널은 32um의 일정한 간격을 유지하고 있으며 병렬로 나란히 배열되어 있다. 스캐닝 동작에서는 한 번에 하나의 센서 채널만 ON 하고, 다음 채널을 ON 하는 동안 이전에 ON 하였던 채널의 데이터가 출력된다. 센서 출력 데이터는 비디오 앰프를 사용하여 증폭을 하고 A/D 변환을 한다.

#### 3.2. 웨이딩 블록

웨이딩 블록은 센서 픽셀 위치에 대한 감도 차이 또는 조도 차이가 발생하더라도 균등한 히스토그램 값을 갖도록 향상 처리를 한다. 히스토그램 향상과정은 센서 픽셀 위치에 대해 화이트 웨이딩과 블랙 웨이딩 값을 사용하여 보정을 수행한다. 이 과정은 식(1)에서 보였듯이 뺄셈과 나눗셈 연산을 반복하게 된다. 많은 인스트럭션 수행을 요구하는 나눗셈 연산은 시스템 초기화시에 LUT로 만들어 놓고 FPGA는 입력되는 센서의 디지털 값에 대한 LUT값을 출력하도록 한다. 웨이딩 블록은 FPGA와 LUT 데이터를 저장하기 위한 RAM으로 구성된다. FPGA는 변환 데이터를 마이크로프로세서 MPU와 동기화된 시퀀스 과정을 거쳐 데이터를 MPU로 전송하도록 하며, MPU의 요청에 대한 ADC 계인 컨트롤을 수행한다.

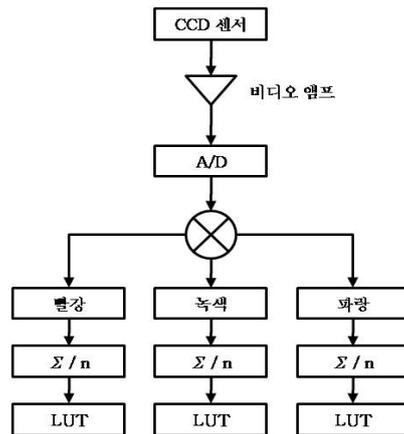


그림 2. LUT 생성 블록도  
Fig. 2 LUT Building Block Diagram

#### 3.3. MPU 블록

MPU 블록은 룩업테이블 생성과 시스템 제어에 관련된 동작을 수행한다. 룩업테이블 생성은 앞에서 설명한 알고리즘을 기반으로 LUT 데이터 계산에 관련한 연산 동작이다. 이 과정은 초기에 한번 수행하고 스캐닝 동작에서는 필요하지 않다. 시스템 제어 동작은 시스템 초기화 과정, 스캐닝 프로세스에 필요한 처리를 의미한다. 시스템 초기화 과정은 시스템 전원이 인가되면 웨이딩 블록의 FPGA에 VHDL 코드를 로딩하고, 생성한 LUT 데이터를 업로드 한다.

#### IV. 실험 및 고찰

제안 알고리즘의 성능을 평가를 위하여 실험은 VHDL을 사용하여 FPGA에 웨이딩 블록을 설계하였다. 설계는 HDL 상향식 설계 기법을 통하여 이루어졌으며 특정 구현 대상에 국한하지 않는 범용적인 설계를 이루고자 IEEE 표준 라이브러리를 사용하였다. 설계된 하드웨어는 Modelsim으로 시뮬레이션 하여 검증 하였다.

표1은 웨이딩 보정 알고리즘 구현을 위한 칩셋의 특성을 보이고 있다. FPGA의 게이트 수는 14,579이고 내부의 블록램은 576이다. 기준 클럭은 66.6Mhz를 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 x 16

그림3은 FPGA와 CCD 센서 사이의 제어 신호타이밍을 나타내고 있다. C\_MCLK는 CCD 센서의 기준 클럭이고, RB는 리셋 게이트 클럭이다. 한 주기는 676ns로 동작한다. A\_CLAM 신호는 센서의 클램프 타이밍이다. 센서의 신호는 A\_DOUT가 'H'로 상승 할 때 출력이 된다.

그림4는 화이트웨이딩 데이터에 대한 3색의 히스토그램을 나타낸 그래프이다. x축 방향은 픽셀의 위치를 상대적으로 나타낸 것이고 y축은 히스토그램의 밝기값을 보이고 있다. 히스토그램의 값은 0~255 범위에 있다. 그림에서 '+' 그래프는 Red 센서의 데이터를 나타내고 있고 '◇'은 Blue를 나타내며 '○'은 Green 이미지의 히스토그램 그래프이다. 조도가 밝은 중앙 부분은 전체적으로 높은 히스토그램 값을 나타내고 있으며 양 끝으로 갈수록 낮은 히스토그램 값을 보인다. 그러나 Red의 경우 0.1~0.3 구간은 편차가 크게 변화하는 현상을 관찰할 수 있다. 이것은 이 부분에서 센서의 개인 또는 렌즈와 같은 외부 조건의 영향이 있음을 알 수 있다. 반면에 동일한 구간에서 Blue와 Green은 완만한 곡선을 보이고 있다. R, G, B 세 채널의 데이터는 픽셀 구간 0.4에서 주변

픽셀의 히스토그램 값보다 낮은 히스토그램 값의 계곡을 보이고 있다. 이것은 조도의 영향으로 3가지 센서 모두에서 발생되고 있음을 알 수 있다.

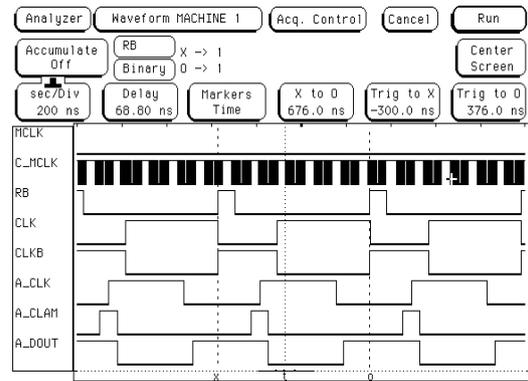


그림 3. 센서 제어 신호 타이밍  
Fig. 3 Sensor Control Signal Timing

그림5는 블랙 웨이딩 데이터의 그래프이다. Blue와 Green은 평균적으로 70~77의 히스토그램 값을 보이고 있다. 반면 Red는 58~65의 히스토그램 값을 갖고 있다. 이것으로 Red 센서는 다른 컬러보다 낮은 감도를 갖고 있음을 알 수 있다. Blue와 Green의 평균 히스토그램 레벨이 비슷한 상태이지만 Red 센서의 평균 히스토그램 레벨이 다른 센서 레벨보다 낮은 경우 컬러 구성의 전체에 왜곡이 발생할 수 있다. 따라서 센서 별로 가중치 값을 적용하여 유사한 레벨의 값을 갖도록 하였다.

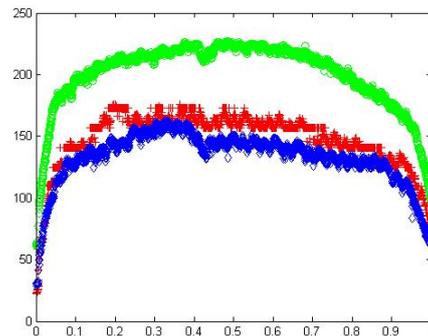


그림 4. 화이트 웨이딩 이미지의 히스토그램  
Fig. 4 Histogram of White Shading Image

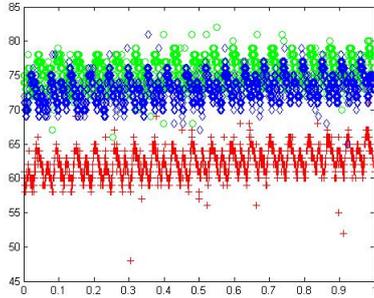


그림 5. 블랙 웨이딩 이미지의 히스토그램  
Fig. 5 Histogram of Black Shading Image

그림6은 LUT 웨이딩 보정 알고리즘을 적용한 후 한 라인의 스캔 이미지 데이터를 나타낸 것이다. 스캐닝 대상은 화이트의 백지를 사용하였다. 그림에서 각 컬러별 평균 히스토그램 값의 편차가 현저하게 줄어들었다는 것을 알 수 있다. Red 센서 데이터의 경우 픽셀 0.35, 0.52, 0.80 위치에서만 평균보다 낮은 히스토그램 값을 보이고 있다.

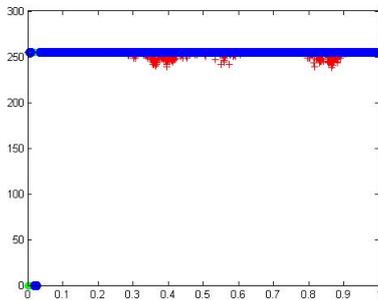


그림 6. 웨이딩 보정 후의 히스토그램  
Fig. 6 Histogram of Shading Correction on White Target

그림7은 Red, Green, Blue로 구성된 패턴 이미지를 스캔 대상으로 하고 LUT 웨이딩 보정 알고리즘을 적용한 스캔 이미지의 한 라인 히스토그램을 그래프로 나타낸 것이다. 보정 전에는 이미지의 양쪽 끝에서 급격한 편차를 보였다.

그러나 보정 알고리즘을 적용한 결과 Blue와 Green 컬러는 완만한 밝기 레벨을 보이고 있다. 입력 패턴의 시작 부분에서 Green에 대한 부분은 히스토그램 그래프의 픽셀 0.45 위치에서 약 50의 밝기 값을 보이고 있다.

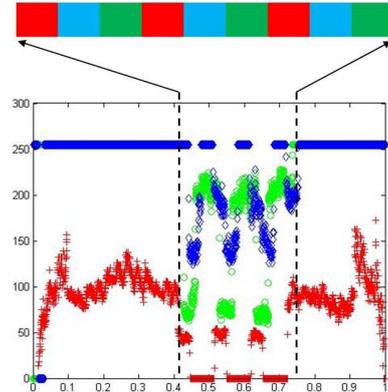


그림 7. 입력 패턴의 히스토그램  
Fig. 7 Histogram of Shading Correction on The Input Pattern

이웃한 Blue 컬러에 대한 스캔 이미지의 히스토그램은 150의 밝기 값을 보였다. Red의 밝기 값은 Blue와 Green에 비하여 상당히 낮은 값을 갖고 있다. Red 패턴의 시작 위치인 픽셀 0.45~0.50 구간에서는 밝기 값이 '0'을 보이고 있고 다른 컬러가 위치한 0.40~0.45 구간에서 '50'의 밝기 값을 보인다. 즉, Red 센서는 자신의 색상이 아닌 다른 컬러에 대해서도 밝기 값의 변화가 있는데 이것은 센서의 특성으로 나타나는 현상이다.

## V. 결 론

본 논문은 CCD 스캔 이미지 향상을 위한 LUT 웨이딩 보정 알고리즘을 이용한 FPGA를 설계 및 구현 하였다. 웨이딩 보정 알고리즘은 연산양 감소를 위해 각 화소에 대해 화이트 및 블랙 웨이딩 조건을 적용하여 센서 출력 값에 대한 모든 값을 LUT화 하였다. 이미지 센서 값에 대한 LUT의 사용은 이미지 처리 과정에서 발생하는 나눗셈과 같은 복잡한 연산을 간소화 할 수 있었고, 저사양의 프로세서를 이용한 스캔 시스템 설계를 가능하게 하였다. 시스템 처리속도는 평균 2.4ms 이하로 안정적이다. 이 결과 범용 프로세서를 사용한 스캐닝 시스템 설계와 저비용의 시스템 구현이 가능할 것으로 기대한다. 앞으로 실용화를 위한 정확한 제품 성능 사양과 임베디드 모듈이 요구된다.

참고문헌

- [1] Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, Prentice-Hall, 2002.
- [2] Kwang Ryol Ryu and Young Bin Kim, "Image Path Searching Auto and Cross Correlations" J. of KIMICS, Vol.9 No.6, pp.747-752, 2011.12.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.

저자소개



**김영빈(Young Bin Kim)**

1993년 목원대학교 전자공학과 공학사  
2000년 목원대학교 전자공학과 공학석사

2011년 목원대학교 전자공학과 공학박사  
2007년~현재 주)디토스 수석연구원  
※관심분야: 영상, 이미지스티칭, 임베디드 시스템



**류광렬(Conan K.R. Ryu)**

1975년 광운대학교 공학사  
1980년 경희대학교 공학석사  
1988년 경희대학교 공학박사  
1996~1997년 University of Pittsburgh  
전기전자공학과 초빙교수

2006~2008년 University of Pittsburgh Medical Center  
뇌신경외과 초빙교수  
현재 목원대학교 전자공학과 정교수  
※관심분야: DSP(음성,영상,비디오,생체), Biomedical Engineering