

논문 2006-43SD-9-6

효율적인 불량화소 검출 알고리즘 및 하드웨어 구현

(An Efficient Dead Pixel Detection Algorithm and VLSI Implementation)

안지훈*, 이원재*, 김재석**

(Jeehoon An, Wonjae Lee, and Jaeseok Kim)

요약

CMOS image sensor는 집적회로 구현이 가능하여 사이즈를 줄일 수 있고 저전력으로 구현이 가능하며 효율적인 영상처리를 할 수 있다는 장점을 갖고 있다. 그러나 불량화소의 발생은 곧 화질의 저하로 연결되기 때문에 불량화소를 검출하는 방법에 대한 연구가 필요하다. 본 논문에서는 CMOS image sensor에 사용되는 효율적인 불량화소 검출 알고리즘과 그 하드웨어를 제안하였다. 불량화소를 검출하기 위하여 본 논문에서 제안한 방법은 Scan, Trace, Detection의 단계를 거친다. 시뮬레이션 결과 특정 조건에서는 99.99%의 불량화소 검출 성공률을 나타냈다. 제안된 알고리즘은 Verilog HDL로 구현 되었으며, 0.25 CMOS standard cell library에서 3.2k개의 게이트 수를 갖는다.

Abstract

In this paper, we propose the efficient dead pixel detection algorithm for CMOS image sensors and its hardware architecture. The CMOS image sensors as image input devices are becoming popular due to the demand for miniaturized, low-power and cost-effective imaging systems. However, the presence of the dead pixels degrade the image quality. To detect the dead pixels, the proposed algorithm is composed of scan, trace and detection step. The experimental results showed that it could detect 99.99% of dead pixels. It was designed in a hardware description language and total logic gate count is 3.2k using 0.25 CMOS standard cell library.

Keywords : Dead pixel, Bad pixel, CMOS image sensor, Image enhancement, detection

I. 서론

최근 CIS(CMOS Image Sensor)는 집적회로 구현이 가능하고 파워를 적게 소비하며 가격이 낮다는 측면에서 각광을 받고 있다. 그러나 CIS에서 불량화소의 발생은 동시에 화질의 열화로 이어지므로 불량화소를 검출하는 방법에 대한 연구가 필요하다^{[1][2]}.

본 논문에서는 불량화소를 검출하는 방법에 대하여 새로운 접근 방법을 제시한다. 기존의 연구에서는 한 화면 내에서 화소를 주변의 화소와 비교하여 불량화소로 의심되는 모든 화소들에 대하여 보상 또는 은닉을

행하여 주는 방법을 주로 사용하였다^[2]. 하지만 이러한 방법은 실제로 불량화소가 아닌 화소에 대하여도 보상 또는 은닉을 행하는 경우가 발생하여 화면을 왜곡시키는 역효과가 있다. 그러나 본 논문에서 제안하는 방법은 한 화면 내에서만 불량화소를 검출하는 것이 아니라 이후의 화면들과도 비교를 행하여 불량화소를 검출하기 때문에 정확한 불량화소의 위치를 검출할 수 있다. 따라서 제대로 동작하는 화소들을 불량화소로 간주하여 왜곡시키지 않는다. 또한 매우 간단한 과정을 거쳐서 불량화소를 검출 할 수 있으므로 하드웨어 구현상의 Overhead가 매우 작다는 장점을 갖고 있다.

본 논문은 다음과 같이 구성되어 있다. II장에서는 불량화소에 관하여 정의하고 불량화소 검출 알고리즘을 제안한다. III장에서는 H/W 구현 결과를 설명하고 IV장에서 결론을 제시한다.

* 학생회원, ** 정회원, 연세대학교 전기전자공학과
(Dept. Electrical and Electronic Eng. Yonsei Univ.)
※ 본 연구는 정보 통신부 및 정보통신연구진흥원 대학 IT연구센터 육성·지원사업의 연구 결과로 수행되었고, CAD Tool은 IDEC으로부터 지원 받았음.
접수일자: 2006년7월1일, 수정완료일: 2006년8월13일

II. 제안하는 불량화소 검출 알고리즘

1. 불량화소의 정의

불량화소의 개수는 CIS의 성능을 결정하는데 중요한 요소이다. 그런데 불량화소의 정의는 아직 확실히 정해진 것이 없다. 일반적으로 불량화소는 다음과 같은 3가지의 종류로 나타난다. (1) 항상 같은 신호를 나타내는 화소, (2) 정해진 임계치보다 큰 잡음을 갖는 화소, (3) 명멸(Blind) 또는 표류(Drift)하는 화소가 그것들이다^[1]. 그림 1은 불량화소의 예를 보여준다.

그러나 일반적인 불량화소의 정의는 범위가 매우 포괄적이기 때문에 검출의 방법으로 사용하기에는 무리가 있다. 따라서 본 논문에서는 다음과 같은 두 가지 특성을 갖는 화소를 불량화소로 정의하고, 불량화소 검출의 방법으로 사용한다.

우리가 영상을 봤을 때 불량화소를 식별해 낼 수 있다는 것은, 주위의 화소와 비교하여서 도저히 나타날 수 없는 화소가 나타났다는 것을 의미 한다. 영상을 주파수 영역에서 보면 낮은 주파수 영역에 많은 에너지를 갖고 있는 것을 확인 할 수 있고, 이것은 영상에서 대부분의 화소들이 이웃한 화소와 비슷한 수준의 화소 값을 갖는다는 것을 나타낸다. 따라서 높은 주파수 성분을 갖는 화소, 즉 주위의 화소들과 비교 했을 때 큰 차이를 갖는 화소는 불량화소일 확률이 크다고 예측할 수 있다.

인간의 눈이 불량화소를 식별할 수 있는 또 다른 이유는 앞에서 설명한, 도저히 나타날 수 없는 화소 값이 같은 위치에서 계속적으로 발생하기 때문이다. 실제로 충분히 움직이고 있는 연속된 화면에서 같은 위치의 화소 값이 k 프레임 뒤에서 거의 변하지 않고 계속 같은 값을 갖는 경우는 굉장히 드문 일이다. 화소 값은 랜덤하게 변화하기 때문에 같은 위치에서 시간 지나도록 같은 값을 나타낸다면 그것은 불량화소일 확률이 높다.

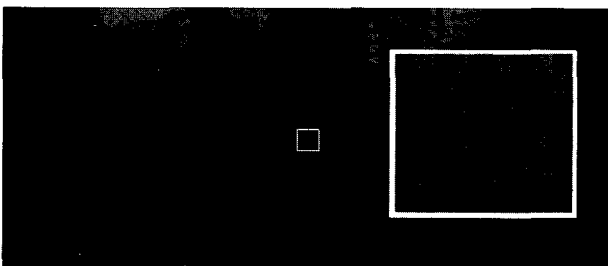


그림 1. 불량화소의 예
Fig. 1. An Example of dead Pixel.

2. 제안된 알고리즘

본 논문에서 제안하는 불량화소 검출 알고리즘의 전체적인 구조는 그림 2와 같다. 그림 3은 제안하는 알고리즘의 순서도를 나타낸다. 검출 방법은 Scan, Trace, Detection의 세 가지 단계를 거친다.

우선 Scan 단계에서는 임의의 시간 t에 발생하는 영상의 전체 화소들에 대하여 Scan을 행한 후, 불량화소로 의심되는 화소의 그룹을 리스트에 저장한다.

이 과정이 끝나면 리스트에 저장된 화소 그룹에 대하여 다음번 화면에서 발생하는 화소 값의 변화를 측정하는 Trace가 행해진다. 비교를 행하는 화면은 시간 t에서 멀리 떨어져 있는 화면일수록 즉, 화면과 상관도가 적을수록 빠르게 검출할 수 있다.

마지막으로 Trace의 과정에서 검출비교회수 N번 비교한 후에도 리스트에 남아있는 화소를 불량화소로 결정한다. 잘못된 검출이 발생하여도 불량화소라고 결정된 화소들에 대하여 Trace과정에서 계속적으로 모니터링을 하기 때문에 다시 수정이 가능하다.

CIS가 오래되어서 발생하는 불량화소는 시간이 지나

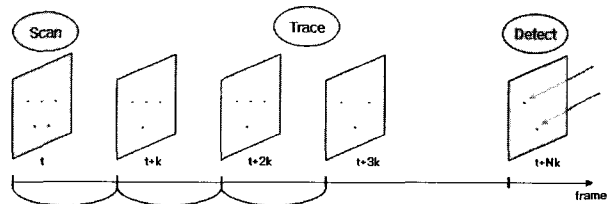


그림 2. 불량화소 검출 알고리즘 진행 방식
Fig. 2. Procedure of the proposed dead pixel detection algorithm.

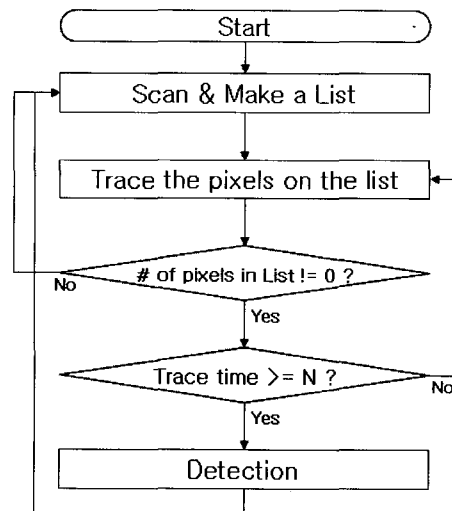


그림 3. 불량화소 검출 알고리즘 순서도
Fig. 3. Flow chart of proposed dead pixel detection algorithm.

면서 점차적으로 발생하므로 검출의 모든 과정이 끝나면 처음의 Scan 단계로 돌아가서 앞의 과정을 반복하게 된다.

가. Scan 과정

앞에서 설명하였듯이 한 화면을 주파수 영역에서 보면 낮은 주파수 성분에 많은 에너지를 갖고 있고, 그것은 낮은 주파수 성분에 많은 정보를 갖고 있음을 의미한다. 따라서 보통의 화소들은 그것을 둘러싼 화소들과 비슷한 레벨의 화소 값을 갖는다. 따라서 주위 화소들의 평균값과 가장 큰 차이를 갖는 화소를 불량화소로 의심할 수 있다.

CIS에서는 RGB의 형태가 Bayer 형태로 입력되기 때문에 각각의 Red, Green, Blue에 대하여 그림 4와 같이 Scan을 행하는 중심화소(x_5 , x 는 R, G 또는 B)와 그것에 이웃하는 8개의 화소들($x_1, x_2, x_3, x_4, x_6, x_7, x_8, x_9$)을 정의한다. 일반적으로 디지털영상처리에서는 처리를 행하는 화소를 중심으로 가로와 세로에 놓인 화소를 사용하여 처리를 수행하지만 본 논문에서는 방향에 관계없이 가장 가까운 곳에 위치하는 화소들을 중심이 되는 화소의 이웃으로 정하였다. 이런 방식을 사용하면 최소 3줄의 영상을 가지고 Scan을 행하여 불량화소로 의심되는 화소들을 정할 수 있다.

8개의 이웃한 화소들과 그것의 중심에 있는 화소 값들과의 차이를 구하기 위해서는 다음과 같은 연산이 필요하다.

Mean of 8-pixels around target pixel:

$$m_{ij} = \frac{\sum_8 x_k}{8} \quad (\text{단, } 1 \leq k \leq 9, k \neq 5) \quad (1)$$

Absolute value of difference:

$$d_{ij} = |P_{ij} - m_{ij}| \quad (2)$$

식 (1)에서 x_k 는 8개의 이웃 화소들을 나타내고 m_{ij}

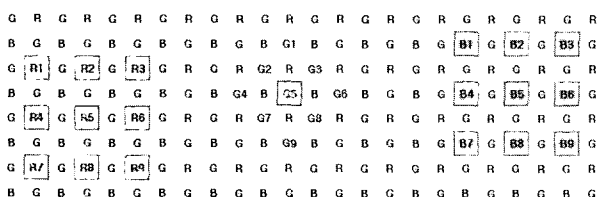


그림 4. 불량화소 검출에 사용되는 화소들
Fig. 4. 9-pixels for dead pixel detection.

는 8개 이웃화소들의 평균을 나타낸다. 식 (2)에서 P_{ij} 는 x_5 즉, Scan을 행하는 중심 화소를 나타내고 d_{ij} 는 이웃 화소들의 평균과 중심 화소의 차이를 나타낸다.

그림 5는 앞에서 설명한 방식으로 한 화면 내의 모든 화소들에 대하여 d_{ij} 값 분포의 예를 나타낸 그래프이다. d_{ij} 값이 1 이하인 점들이 가장 많이 존재하고, d_{ij} 값이 큰 점들은 거의 존재 하지 않음을 알 수 있다.

그림 6은 임의의 화면에서 모든 픽셀들의 d_{ij} 값을 내림차순으로 정렬하였을 경우, 불량화소가 몇 번째로 큰 d_{ij} 값을 갖는지의 예를 나타낸 것이다. 각각의 화면에서 d_{ij} 값이 가장 큰 10개 화소 내에 dead pixel이 존재할 확률이 50%가 넘는다는 것을 알 수 있다. 따라서 본 논문에서는 위의 방법으로 계산한 d_{ij} 값이 가장 큰 순서대로 화소를 10개 추출하고 화소의 위치와 값을 리스트에 저장하여 실험을 진행한다.

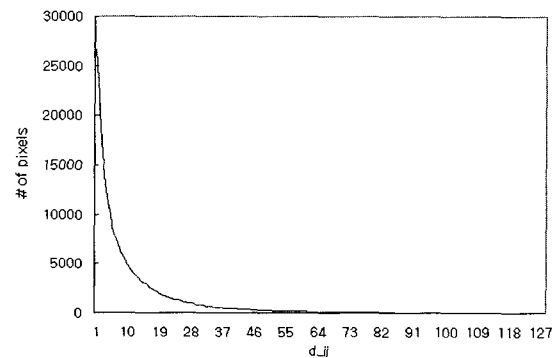


그림 5. 한 화면 내 화소들의 d_{ij} 값 분포의 예
Fig. 5. An example of d_{ij} 's distribution on each pixels in a picture.

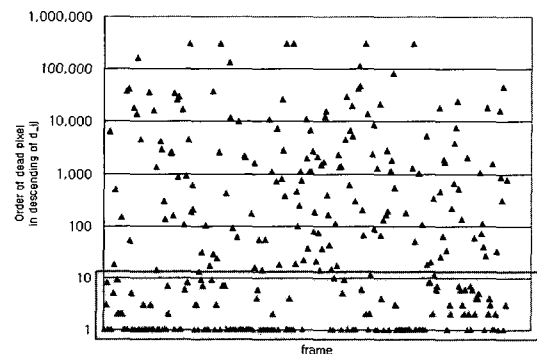


그림 6. 각각의 화면에서 모든 픽셀을 d_{ij} 값에 대하여 내림차순으로 정렬한 경우 불량화소가 나타나는 순위들의 예
Fig. 6. Orders of dead pixel in descending of d_{ij} values in each pictures.

나. Trace 및 Detection 과정

어떤 화소가 불량화소라고 판단하는 가장 큰 기준은 화소 값이 시간이 지남에 따라서도 거의 변하지 않는다는 것이다. 불량화소가 존재 할 때, 앞에서 작성한 리스트에 존재하는 n개의 화소들을 다음번 영상의 화소와 비교하여 값이 정상적으로 변하는 화소들을 지워나가면 값이 변하지 않는 화소를 쉽게 구해낼 수 있다.

화면간 거리가 클수록 영상간의 상관도가 작아지기 때문에 리스트에 저장되어 있는 화소 값들을 바로 다음번 화면의 화소 값과 비교하지 않고 k번 후의 화면의 같은 위치의 화소를 비교한다. 실제로 Scan 과정의 결과로 나온 리스트에 저장되어 있는 화소를 바로 다음번 화면의 화소와 비교했을 때 보다 k 프레임 후의 화면의 화소와 비교했을 때 더 적은 비교 과정을 거쳐서 판단을 할 수 있다. 그림 7은 화면 간 거리에 따른 판별 가능 비교 횟수의 예를 나타낸다.

그림 8은 본 논문에서 제안된 알고리즘을 구현하기 위하여 사용되는 리스트의 예를 나타낸 것이다. 우선 Scan과정에서는 불량화소로 의심되는 n개의 화소들의 위치와 값을 리스트에 저장한다. Trace과정에서는 리스트에 저장되어 있는 값과 현재의 화소 값의 변화를

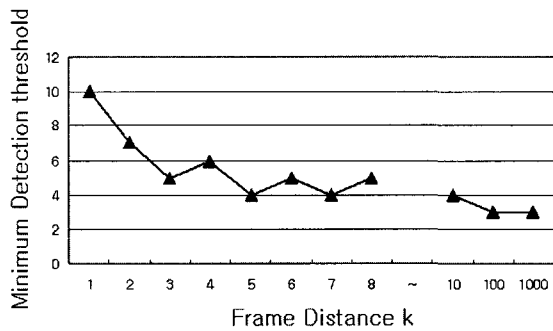


그림 7. 화면간 거리에 따른 최소 비교 횟수의 예
Fig. 7. Minimum detection threshold according to Frame distance.

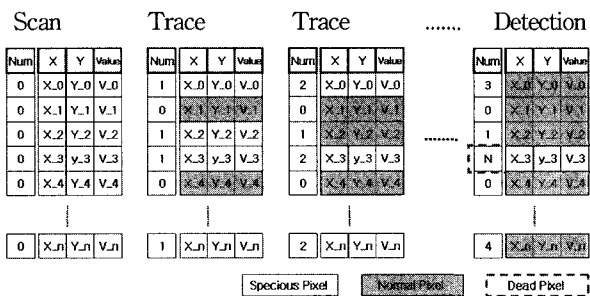


그림 8. 리스트 사용의 예
Fig. 8. An example of list usage.

비교하여 정상적으로 동작하는 화소일 경우 리스트에서 그 화소의 정보를 삭제한다. Trace과정에서 N번 비교 후에도 리스트에 남아있는 화소를 불량화소로 검출한다.

3. 실험

표 1과 같은 조건으로 시뮬레이션을 수행하였다. 테스트 영상으로는 일반 영상으로부터 Bayer 형태의 연속된 영상 시퀀스를 추출하여 사용하였다. 또한 불량화소는 시뮬레이션 시작과 동시에 임의의 값으로 임의의 위치에 발생하여 시뮬레이션 하는 동안에 큰 값의 변화 없이 약간의 진동만 하는 것으로 모델링하였다.

각각의 조건에 대하여 1000회 이상의 시뮬레이션을 수행하였고 각각의 경우 시뮬레이션 시작 화면 즉, Scan을 행하는 화면은 랜덤하게 정하고 그 뒤에 나오는 연속된 화면들 속의 픽셀들에 대하여 Trace 및 Detection을 수행하였다. 그리고 앞의 그림 6에서 설명하였듯이 리스트의 크기를 10으로 고정하여 시뮬레이션을 수행하였다. 또한 리스트의 크기를 증가시키는 경우, 하드웨어가 커지고 복잡도가 증가함에 따라 얻게 되는 속도의 이득은 매우 작을 것으로 판단된다.

표 2는 각각의 조건에 대하여 Detection이 발생하였을 때, 불량화소를 정확히 Detection한 경우를 백분율로 나타낸 것이다. 비교하는 화면들 사이의 간격 k가 클수록, 비교회수 N이 클수록 검출 성능이 더 좋아짐을 알 수 있다. 또한 앞에서 설명했듯이 잘못된 Detection이 발생하더라도 불량화소로 검출된 값들에 대한 지속적인

표 1. 실험 조건

Table 1. Experimental conditions.

List size	10	
Frame distance k	5, 10, 15, 20	
Detection threshold N	10, 15, 20, 25, 30, 35	
Dead pixel	location	random
	value	random
	variation	± 5

표 2. 실험 결과

Table 2. Experimental results.

(단위: %)

k	N	Detection threshold					
		10	15	20	25	30	35
Frame distance	5	73.91	90.64	89.97	93.98	95.99	94.65
	10	91.30	93.65	97.32	95.32	97.67	98.34
	15	91.97	94.98	97.34	98.31	99.67	99.99
	20	92.98	97.63	99.35	99.99	99.99	99.99

검사를 수행할 수 있으므로 다시 수정하는 것이 가능하다. 그러나 화면간의 간격 k 와 비교회수 N 이 큰 경우, 불량화소를 검출하는데 필요한 시간이 증가하게 되므로 적절한 값을 선택할 필요가 있다.

III. 하드웨어 구현

본 논문에서 제시한 알고리즘은 그림 9와 같은 구조로 설계되었고, $0.25\mu\text{m}$ CMOS 공정으로 합성했을 때 $3.2k$ 의 복잡도를 갖는다.

(1) 입력되는 화소 값은 Color Interpolation 블록 내부의 버퍼에 있는 화소 값을 사용한다. a에는 8-bit 중심 화소가 입력되고 b로는 8-bit의 이웃 화소 8개가 병렬로 입력된다. c에는 Color Interpolation 블록의 카운터로부터 중심화소의 좌표가 입력된다.

(2) b에 입력된 8개의 이웃 화소들의 평균을 구하고 a로 입력된 중심화소와의 차를 계산하여 절대값을 취하게 되고, 이 값은 a로 입력된 중심 화소와 함께 MUX의 입력으로 들어온다.

(3) 카운터의 값이 0일 때는 Scan을 행하고, Trace의 과정에서 비교할 때마다 카운터의 값을 1씩 증가한다. 카운터의 값이 검출 비교회수 N 에 도달하면 Detection을 수행한다. 만약 리스트가 비면 카운터의 값을 0으로 만들고 다시 Scan을 행하게 된다.

(4) MUX는 카운터 값이 0이면 Scan을 행하게 되므로 d_{ij} 를 내보내고 그 외의 값에서는 Trace를 행하므로 중심 화소의 현재 값 P_{ij} 를 출력으로 내보낸다.

(5) Controller는 Scan 과정에서 저장되어 있는 화소

들 중에서 가장 작은 d_{ij} 값을 갖는 화소를 알고 있고 이후에 들어오는 화소의 d_{ij} 값이 저장되어있는 화소들 중에서 가장 작은 d_{ij} 보다 큰 경우에 가장 작은 d_{ij} 값을 갖는 화소를 버리고 새로운 화소를 리스트에 저장한다. 이런 방식으로 가장 큰 d_{ij} 값을 갖는 10개 화소들의 위치와 값을 리스트에 저장한다. Trace과정에서는 리스트를 더 이상 업데이트 하지 않고 리스트에 저장되어 있는 화소 값과 현재 화면의 픽셀 값 P_{ij} 값을 비교하여 제대로 동작하는 화소가 맞을 경우에는 리스트에 들어있는 화소 정보를 버린다. 카운터의 값이 검출 비교회수 N 에 도달하면 Detection을 행하여 리스트에 남아있는 화소들을 불량화소로 검출한다.

(6) 리스트에는 픽셀의 위치와 값이 저장된다. 리스트를 이루는 메모리는 픽셀의 위치와 값이 각각 2byte, 1byte가 필요하고 리스트의 크기를 10으로 하였기 때문에 전체 30byte를 저장할 레지스터가 필요하다.

(7) 결과 값으로 Color Interpolation 블록에 불량화소의 위치를 전달하고, Color Interpolation 블록에서는 불량화소에 대하여 은닉을 행한다.

IV. 결론

본 논문에서는 불량화소를 검출하는 알고리즘을 제시하였다. 시뮬레이션 결과 일정 조건에서 99.99%의 높은 불량화소 검출률을 보였다. 불량화소의 또한 하드웨어 구현을 고려한 간단한 구조로 되어있기 때문에 $3.2k$ 개 게이트의 낮은 복잡도를 갖는다. 또한 본 논문에서 제안한 방법은 불량화소의 정확한 위치를 찾아내기 때문에 불량화소가 아닌 화소에 대하여 행해지는 은닉으로 인하여 발생하는 화질의 저하와 같은 역효과를 줄일 수 있다는 장점이 있다.

참고 문헌

[1] José Manuel López-Alonso, Javier Alda, "Bad pixel identification by means of principal components analysis," *Optical Engineering*, Vol. 41, No. 9, pp. 2152-2157, September 2002.

[2] Xu Youqing, Yu Shengsheng, Zhou Jingli, Fang Zuyuan "Detection and Compensation of Bad Pixel for CMOS Image Sensor," *International Conference on Sensor and Control Techniques (ICSC 2000) Proceedings of SPIE*, Vol. 4077, pp. 208-212, Wuhan, China, June 2000.

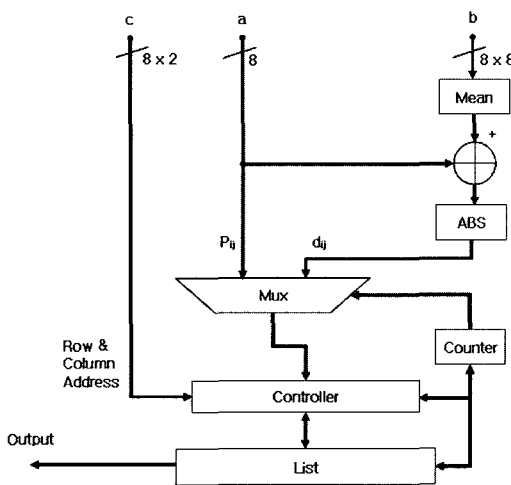


그림 9. 제안된 알고리즘의 H/W 블록
Fig. 9. H/W block of proposed algorithm.

저 자 소 개



안 지 훈(학생회원)
 2006년 연세대 전기전자공학과
 학사.
 2006년~현재 연세대 전기전자
 공학과 석사 과정.
 <주관심분야 : 영상처리, SoC 설
 계>



이 원 재(학생회원)
 2001년 연세대 전기전자공학과
 학사 졸업.
 2003년 연세대 전기전자공학과
 석사 졸업.
 2003년~현재 연세대 전기전자
 공학과 박사 과정.
 <주관심분야 : 영상처리, SoC 설계>



김 재 석(정회원)
 1977년 연세대 전자공학과 학사
 졸업.
 1979년 KAIST 전기전자공학과
 석사 졸업.
 1988년 Rensselaer Polytechnic
 Institute, NY, 박사 졸업.
 1993년~1995년 한국전자통신연구원 책임연구원
 1996년~현재 연세대학교 전기전자공학과 교수
 <주관심분야 : 통신 및 영상 시스템, VLSI 신호
 처리, 임베디드 S/W 및 SoC 구현>