CMOS Image sensor 를 위한 효과적인 플리커 검출기 설계

이평우 이정국 김채성
매그나칩 반도체 ISD CIS 개발팀

Design of Efficient Flicker Detector for CMOS Image Sensor

Pyeongwoo Lee, Jeongguk Lee, Chaesung Kim
CIS Development team, Imaging Solution Division
Magnachip Semiconductor Ltd.
E-mail: {pyeongwoo.lee, jeongguk.lee, chaesung.kim}@magnachip.com

Abstract

In this paper, an efficient detection algorithm for the flicker, which is caused by mismatching between light frequency and exposure time at CMOS image sensor (CIS), is proposed.

The flicker detection can be implemented by specific hardware or complex signal processing logic. However it is difficult to implement on single chip image sensor, which has pixel, CDS, ADC, and ISP on a die, because of limited die area. Thus for the flicker detection, the simple algorithm and high accuracy should be achieved on single chip image sensor,.

To satisfy these purposes, the proposed algorithm organizes only simple operation, which calculates the subtraction of horizontal luminance mean between continuous two frames. This algorithm was verified with MATLAB and Xilinx FPGA, and it is implemented with Magnachip 0.18 standard cell library. As a result, the accuracy is 95% in average on FPGA emulation and the consumed gate count is about 7,500 gates (@40MHz) for implementation using Magnachip 0.18 process.

I. 서론

CMOS Image Sensor 는 회로 집적 능력과 전력 소모 면에서 CCD 방식의 이미지 센서보다 우수한 특성을 가지기 때문에, 최근 Digital Still Camera 부터 카메라 폰의 image sensor 까지 널리 사용되고 있다[1]. 또한 CMOS Image Sensor 는 Rolling-shutter 방식을 사용하여 픽셀의 라인 별 노출 시간 할당과 CDS(Correlated Double Sampling)를 처리함으로 빠른 영상 출력 속도, 저잡음, 저전력 동작이 가능한 이점을 가지고 있다[2][3].

이러한 Rolling-shutter 방식은 형광등과 같이 발산하는 빛의 량이 시간에 따라 주기를 가지고 변하는 광원과 함께 촬상 되는 경우, 각 라인에 대해 픽셀 노출 시간을 동일하게 할당하여도 라인당 수광하는 빛의 량이 상이하게 된다. 따라서 실제 촬상된 화면에는 각 다른 휘도 값을 가지는 특정한 띠가 주기적으로 나타나게 된다. 이와 같이 원 영상에 부가되어 나타나는 띠를 플리커(Flicker)라 하며, 광원의 주기를 고려하여 픽셀 노출 시간을 조정함으로 플리커를 제거하기 위한 연구가 있었다[4]. 하지만 자동 노출 장치(Auto Exposure)가 광원의 주파수 에 대한 정보를 가지고 있어야 이를 고려한 노출 시간 할당이 가능한 단점이 있어, 잘못된 광원의 주파수 정보가 사용되는 경우 촬상 이미지에 플리커가 나타나는 현상을 가져올 수 있다.

상기와 같은 문제를 막기 위해, 실제 광원 주파수와 다른 값이 사용되었을 때 화면에 생성되는 플리커를 감지하는 플리커 검출 회로가 요구된다. 플리커 검출을 위한 방법으로는 별도의 수광부를 가지는 Super pixel 을 사용하는 방법[5]과 복잡한 신호처리를 사용하는 방법이 있으나, Super pixel 을 사용하는 방법은 별도의 아날

로그 회로를 요구하며, Sensitivity 가 높은 특수한 픽셀 설계를 필요로 하기 때문에 실제 적용이 어려운 단점이 있다. 또한 복잡한 신호처리를 사용하는 방법은 제한된 면적(Die Size)을 사용하는 Image sensor 에 집적 시, 구현 면적상의 제약으로 많은 하드웨어 자원을 할당하여 설계하기 어려운 문제가 있어 실제 적용에 부담을 주게 된다.

따라서 본 논문에서는 적은 하드웨어 자원으로 효율적인 플리커 검출이 가능한 알고리즘을 제안하고, MATLAB 모의 실험과 FPGA 검증을 통해 실제 구현시의 성능을 평가하여, CMOS Image Sensor 와 단일 칩으로 집적되는 ISP(Image Signal Processor)를 위한 플리커 검출기를 설계하였다.

II. 플리커 검출 알고리즘

본 논문에서 제안하는 알고리즘은 이미지 센서에서 얻어진 영상에서 플리커를 추출하는 방식을 구현함에 있어 순수 획득 영상에서 신호 처리만을 기반으로 하기 때문에 이미지 센서 제작 공정의 영향을 받지 않고 사용할 수 있는 이점이 있다. 제안하는 알고리즘은 다음과 같은 순서로 플리커를 검출하게 되며 그림 1 에 이를 도시하였다.

첫째, 각 영상 데이터로부터 라인 별 평균 휘도 값을 계산한다. 둘째, 계산된 평균 휘도 값을 통해 현재 프레임의 데이터로부터 이전 프레임의 데이터의 차를 구하여 플리커 곡선(Flicker curve)을 추출한다. 셋째, 얻어진 차 값을 저역 통과 필터(Low pass filter)를 통과시켜서 불필요한 국소점 들을 제거한다. 넷째, 이렇게 얻어진 데이터에서 골(trough)을 파악하여 그 주기를 계산함으로 실제 플리커에 의해 일어난 것인지를 판단하여 플리커가 생겼는지 여부를 판별하게 된다.

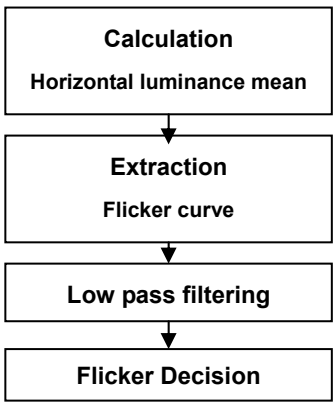


그림 1 플리커 검출 알고리즘

2.1 라인 별 평균 휘도 값 계산

본 논문에서 제안하는 알고리즘은 연속하는 두 영상에서 추출한 차 영상을 기반으로 플리커를 검출한다. 하지만 연속하는 두 영상에서 차 영상을 구하기 위해서는 각 영상 전체를 저장할 수 있는 프레임 메모리를 사용해야 함으로 프레임 메모리로 인해 구현 면적이 매우 많이 소요되게 된다. 이러한 문제를 해결하기 위해 그림 2 와 같이 가로 방향 라인 별 평균 휘도 값을 계산하여 대표 값으로 사용한다.

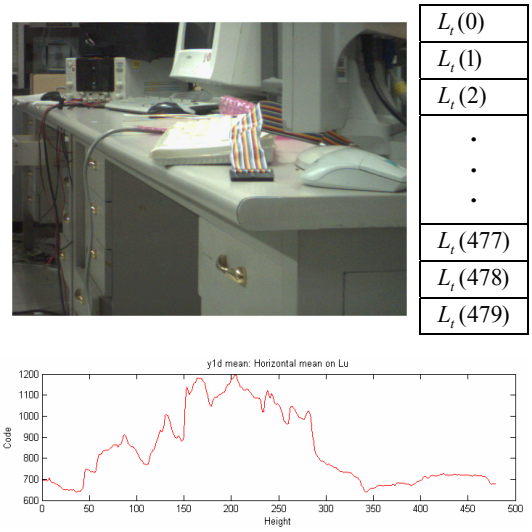


그림 2 라인 별 평균 휘도 값 계산의 예

여기서 라인 별 평균 휘도 값은 각 영상에 따라 아래 수식 2 와 수식 3 을 통해 계산되며, 이는 수식 1 에 기반하고 있어 계산된 라인 별 평균 휘도 값은 피사체(Object)와 플리커(Flicker)에 대한 정보를 모두 포함하게 된다. 수식 4 는 현재 영상 정보에서 이전 영상 정보와의 차를 계산하는 과정으로 피사체에 대한 정보를 상쇄하고, 플리커의 이동에 따른 잔상 정보만을 $S_t(h)$ 로 포함하게 되며 이를 플리커 곡선(Flicker curve)이라 한다.

$$Lu_t = Luminance\ (Object + Flicker) \quad \text{(수식 1)}$$

$$L_{t-1}(h) = \left(\sum_{i=1}^{width} Lu_{t-1}(i) \right) / width \quad \text{(수식 2)}$$

$$L_t(h) = \left(\sum_{i=1}^{width} Lu_t(i) \right) / width \quad \text{(수식 3)}$$

$$S_t(h) = \left| L_{t-1}(h) - L_{t-1}(h) \right| \quad \text{(수식 4)}$$

Lu_t; *Luminance at time t*

$L_t(h)$; *mean of Lu. on height h, at time t*

$S_t(h)$; *Lu.difference between time t and* $t-1$

2.2 플리커 곡선의 추출과 저역 통과 필터의 실행

상기 과정을 통해 얻은 플리커 곡선은 이미지 센서의 잡음 성분과 실제 하드웨어 회로에서 사용되는 정수 연산의 성질 때문에 그림 3 과 같이 많은 골(trough)를 가지게 된다.

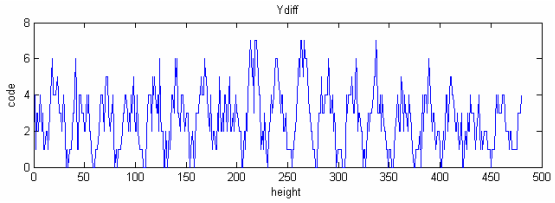


그림 3 플리커 곡선의 예

간단한 회로만을 사용하여 플리커 곡선의 주기성을 판단함에 있어 잘못된 골 때문에 주기성이 왜곡될 수 있다. 따라서 저역 통과 필터(Low-pass filter)를 실행하도록 하여 에너지가 큰 골만을 남기고 나머지는 제거하게 된다. 그림 4 는 그림 3 의 플리커 곡선을 저역 통과 필터를 실행한 결과를 나타낸다.

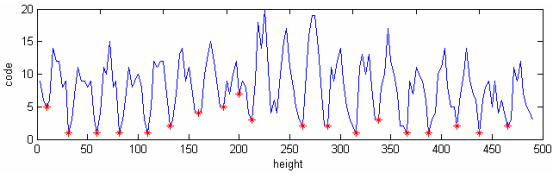
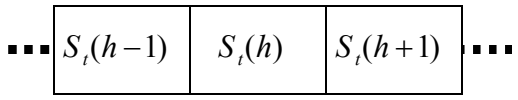


그림 4 저역 통과 필터의 실행 결과

2.4 플리커의 판단

저역 통과 필터를 통과한 플리커 곡선의 주기성을 판단하기 위해 그림 5 와 같이 기준 값 전후의 값을 비교하여 기준 값이 전후의 값보다 작다면 골로 판단하고 골의 위치를 기억한다.



$$T_t(h)=\begin{cases} 0 & (S_t(h-1)\geq S_t(h)\ or\ S_t(h)\leq S_t(h+1)) \\ h & (S_t(h-1)<S_t(h)\ and\ S_t(h)<S_t(h+1)) \end{cases}$$

그림 5 골(trough)의 판단

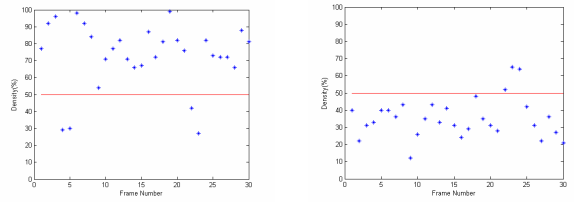
그 후, 다음 골이 나타날 때 이전 골과의 거리를 계산하여 골의 주기성을 판단하게 된다. 하지만 저역 통과 필터를 통과했어도 그림 4 와 같이 일부 지역적으로 나타나는 작은 골을 모두 제거하지 못함으로 그림 5 의 방법에 의해 골로 판단되더라도 이전 골과 거리가 임의 값보다 작은 경우, 골로 판단하지 않는 처리가 필요하다. 그림 4 의 * 표시는 이러한 처리를 통해 큰 에너지를 갖는 골만을 검출해 놓은 결과이며, 골과 골 사이의 거리가 짧은 일부 부분에 대해서는 골로 인정되지 않아 * 표시가 되지 않았음을 나타낸다.

이와 같은 과정을 통해 검출된 골이 일정한 거리마다 규칙으로 나타나는 빈도(Flicker density)가 임의의 값보다 큰 경우, 플리커 성분으로 인한 골으로 인지하고 현재 영상에 플리커가 존재함을 자동 노출 장치(Auto-Exposure)에게 알린다. 이를 통해 자동 노출 장치는 50Hz 광원용 노출 조정 모드에서 60Hz 광원용 모드로 변경하거나, 그 반대로 변경함으로 현재 영상에서 플리커를 제거한다.

III. MATLAB 모의 실험 결과

본 논문에서 제안한 플리커 검증 알고리즘의 설계에 들어가기 앞서, MATLAB 으로 모의 실험해 본 결과, 영상에 플리커가 있는 경우와 없는 경우에 대해 그림 6 과 같은 결과를 나타내었다.

모의 실험을 위해 연속된 30 장의 VGA 영상을 사용하였으며, 실제 영상의 연속 저장을 위해 JPEG 포맷을 사용하였다.



(a) Flicker image (b) Non-flicker image

그림 6 모의 실험 결과

모의 실험 결과, 그림 6 (a)과 같이 Flicker Density 의 문턱 값(threshold)을 50%로 설정했을 때 플리커가 있는 영상의 경우, 30 장의 영상 중 26 장의 영상을 플리커 영상으로 판단하여 약 86.6%의 정확도를 보였다.

또한 플리커가 없는 영상의 경우, 그림 6 (b)와 같이 30 장의 영상 중 27 장의 영상을 플리커 없는 영상으로 판단하여 90%의 정확도를 보여, 비교적 높은 정확도를 나타내었다.

IV. 하드웨어 설계 구현

제안된 플리커 검출 알고리즘을 하드웨어로 설계하고 이를 검증하기 위해 FPGA Prototyping Platform 을 제작하여 구동하였으며, 검증된 IP 는 매그나칩 0.18um Standard Cell 로 구현되었다.

4.1 FPGA Prototyping

그림 7 과 같은 FPGA 구성이 사용되었다.

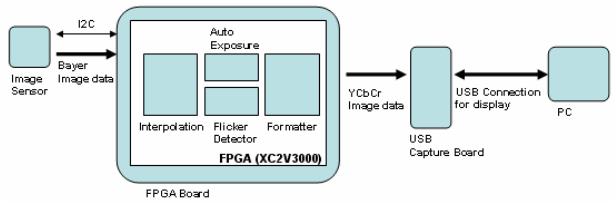


그림 7 FPGA Prototyping Platform

설계된 플리커 검출기의 검증을 위해 Interpolation, Auto Exposure, Formatter 와 함께 Xilinx XC2V3000E 에 집적하였으며, Image sensor 는 매그나칩 반도체의 CMOS Image Sensor(CIS)를 사용하였다.

검증 결과, 플리커 있는 영상이 촬상된 경우 30 회의 시도 중, 플리커가 존재한다는 판단을 29 회 성공하여 약 97%의 검출 정확도를 보였으며, 자동 노출 장치(Auto Exposure)로 하여금 픽셀 노출 시간 보정을 가능하게 하였다. 또한 플리커 없는 영상이 촬상된 경우 30 회의 시도 중 플리커 없다는 판단을 28 회 성공하여 93%의 정확도를 나타내어, 플리커가 없는 경우 자동 노출 장치의 픽셀 노출 시간 보정이 일어나지 않아, 안정적인 동작이 가능하였다.

4.2 설계 결과

FPGA 로 검증된 플리커 검출기를 CMOS Image sensor 의 ISP(Image Signal Processor)와 함께 집적하기 위하여, Synopsys 의 Physical compiler 와 매그나칩 반도체 0.18um 공정을 이용하여 설계한 결과, 약 7,500 gate(@40MHz)가 소요되었으며, 차영상 계산을 위한 라인별 평균 휘도 값 저장을 위해 2KB 의 SRAM 이 사용되었다.

V. 결론

본 논문에서는 CMOS Image Sensor 에서 각 라인별로 상이한 광원을 입력으로 받을 경우 각 라인이 다른 휘도 값을 가지게 됨으로 발생하는 플리커 현상을 감지하기 위한 플리커 검출기 알고리즘을 제안하고 구현하였다. 제안한 방법은 연속 영상의 차영상을 통해 피사체의 정보를 제거한 후, 플리커 곡선을 추출하여 그 주기성을 판단한다.

실험 결과 제안된 플리커 검출 방법은 플리커를 검출하여 자동 노출 장치로 하여금 노출 시간을 보정하여 플리커를 제거하는 데 적합하다는 것을 알 수 있었다. 향후 과제로서 현재 플리커 검출기는 약 7,500gate 의 적은 하드웨어 자원으로 설계가 가능하였으나, 차영상을 얻기 위해 2KB 의 SRAM 을 사용하였다. 따라서 SRAM 의 소요량을 줄이기 위한 최적화 작업이 진행되야 할 것이다.

참고문헌

[1] S. G. Wuu et al., “A High Performance Active Pixel Sensor with 0.18-um CMOS Color Imager Technology," IEEE IEDM Technical Digest, pp.555-558, December 2001.

[2] A. Fish, E. Avner and O. Yadid-Pecht, “Low-Power global/Rolling Shutter Image Sensors in Silicon on Sapphire Technology,” Proceeding of IEEE ISCAS, May 2005.

[3] N. Stevanovic, M. Hillegrand, B. J. Hostica and A.Teuner, “A CMOS Image Sensor for High Speed Imaging,” Proceeding of IEEE ISSCC, pp. 104-105, February 2000.

[4] N. Masaru, T. Kinugasa, US Patent 4774588, 1987.

[5] J. Hurwitz et al., “A Miniature Imaging Module for Mobile Applications,” IEEE ISSCC Digest of Technical Papers, pp. 90-91, Feb. 2001