

ISP를 위한 효율적인 Color Interpolation 알고리즘

정영식 방정배⁰ 장영조
한국기술교육대학교

sikiga@hanmail.net

jbbang@kut.ac.kr

yjjang@kut.ac.kr

Efficient Algorithms on Color Interpolation for ISP

Jeung Young Sic, Bang Jeong Bae⁰, Jang Young Jo
Korea University of Technology and Education

요약

영상을 동반한 통신이 일반화되면서 CMOS 이미지센서를 장착한 저렴한 영상장치의 수요가 증대되었다. 하지만 CMOS 이미지 센서는 CCD센서에 비해 화질이 좋지 않으므로 이를 개선해주는 전처리단계의 중요성이 더욱 부각되었다. 본 논문에서는 CMOS 이미지 센서에 의해 포획된 동영상을 실시간으로 처리하여 화질을 개선하는 ISP를 구현하고 시뮬레이션하였다. ISP의 여러 단계중 Color Interpolation에 중점을 두어 여러 알고리즘을 보이고 비교/분석하여 각 상황에 맞는 알고리즘을 제시하였다.

1. 서론

단순한 문자나 음성 통신에서 영상을 동반하는 통신이 일반화되고 있다. 개인 컴퓨터나 휴대폰 등에서 영상의 입력 장치로 비용, 소비 전력 등의 측면에서 CCD 이미지센서보다 유리한 CMOS 이미지센서를 사용한 카메라가 급속히 확산되고 있다. 그러나 CMOS 이미지센서는 CCD 이미지센서에 비해 화질이 좋지 않으므로 이를 개선해주는 전처리 단계인 ISP (Image Signal Processor)가 중요한 역할을 한다.

ISP의 과정은 그림 1에 보인 블록도에 의하면 입력영상의 불충분한 영상 데이터를 기반으로 온전한 영상 데이터를 생성하는 Color Interpolation, 입력된 영상의 3가지 색(빨강:R, 파랑:B, 초록:G)을 자연의 색에 맞게 변화시키는 White Balance, 각 픽셀들의 값이 R, G, B로 구성되어 있는 것을 Y, Cb, Cr로 바꾸는 Space Conversion, 입력된 영상과 디스플레이의 차이를 보정하기 위한 감마 보정 (gamma correction), 입력된 영상의 잡음을 제거하고 영상을 선명하게 하기 위한 필터처리 등이 있다.

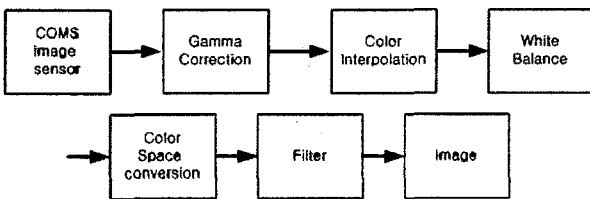


그림 1 ISP의 대략적인 블록도

여기에서 Color Interpolation은 CMOS 영상 센서에서 출력되는 영상 데이터를 그림 2와 같은 Bayer pattern 형태의 불충분한 값을 토대로 하여 원 영상과 비슷한 영상으로 나타내기 위해 수행하는 과정이다. 즉, CMOS 영상 센서의 값을 가지고 초기의 영상을 내삽하는 과정으로 다른 과정들에게 미치는 영향이 매우 크다. 본 논문에서

는 CMOS 영상 센서 ISP를 위한 여러 가지 처리과정 중에서 Color Interpolation을 수행하기 위한 몇 가지 방법을 보이고 비교/분석하여 ISP의 용도에 따른 적합한 알고리즘을 제시하고 그 유효성을 실험으로 확인하고자 한다.

R	G	R	G
G	B	G	B
R	G	R	G
G	B	G	B

그림 2. Bayer pattern

II. 본론

2.1 2*2 매트릭스 알고리즘

이 알고리즘은 가장 기본적인 것으로 그림 3과 같이 4개의 이웃하는 입력 데이터를 받아 그에 상응하는 픽셀들을 완전한 영상 데이터(R,G,B)로 계산한다. 블록에는 R과 B가 하나씩, G값은 2개가 포함된다. (하나의 매트릭스단위를 블록이라 한다.) 그래서 식 (1)과 식(3)에 의해서 한 블록의 모든 R 또는 B의 값은 동일하며, 식(2)으로 인해 G는 최대 3개의 각기 다른 값을 지니게 된다.

		R	G	R	G
		G	B	G	B
R	G	R	G	R	G

그림 3. 2*2 매트릭스 알고리즘

$$P_{1R} = R \quad (1)$$

$$P_{1G} = (G_1 + G_2) / 2 \quad (2)$$

$$P_{1B} = B \quad (3)$$

2.2 3*3 매트릭스 알고리즘

9개의 이웃하는 입력 데이터를 이용하여 그에 상응하는 영상 데이터를 계산한다. 그래서 그림 4와 같이 한 블록에 포함되는 R과 B의 수가 홀수 번째의 블록과 짝수 번째의 블록들 사이에는 유효값의 수가 다르게 포함되어 계산식도 각각 다르게 표현된다. 픽셀 데이터를 구하는 수식 중 아래와 같다.



(a) (b)
그림 4. 3*3 매트릭스의 예

(a)의 경우

$$R = (R_1+R_2+R_3+R_4)/4 \quad (4)$$

$$G = (G_1+G_2+G_3+G_4)/4 \quad (5)$$

$$B = B \quad (6)$$

(b)의 경우

$$R = (R_1+R_2)/2 \quad (7)$$

$$G = G_3 \quad (8)$$

$$B = (B_1+B_2)/2 \quad (9)$$

2.3 3*6 매트릭스 알고리즘

그림 5와 같이 18개의 이웃하는 입력 데이터를 이용하여 그에 해당하는 픽셀 데이터를 계산하는 알고리즘이다. 3*3 매트릭스를 이용한 것보다 색상을 원 영상에 가깝게 표현하게 되고 모든 색에 대하여 모든 블록의 유효값들의 수가 일정하다. 아래의 수식은 R의 경우를 예로 들었다.

	p1	p2	p3	p4	p5	p6
	p7	p8	p9	p10	p11	p12
	p13	p14	p15	p16	p17	p18

그림 5. 3*6 매트릭스 알고리즘

$$R_{12}=(p1+p3)/2 \quad (10)$$

$$R_{22}=(p1+p3+p13+p15)/4 \quad (11)$$

2.4 개선된 2*2 매트릭스 알고리즘

기존의 2*2매트릭스 알고리즘에서 R과 B를 개선한 것으로 그림 6과 같이 2*2매트릭스에 R과 B를 위한 1*3필터를 붙인 구조를 지닌다. G를 계산하는 수식은 종전과 같고 R과 B의 경우 1*3 필터에 의해 수식(12)과 수식(14)와 같이 된다.

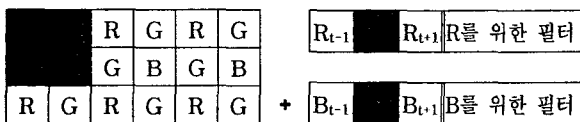


그림 6. 개선된 2*2 매트릭스 알고리즘

$$P_{1R} = \begin{cases} R & (R_{t-1} = R_{t+1}) \\ (R_{t-1} + R_{t+1})/2 & (R_{t-1} \neq R_{t+1}) \end{cases} \quad (12)$$

$$P_{1G} = (G_1+G_2)/2 \quad (13)$$

$$P_{1B} = \begin{cases} B & (B_{t-1} = B_{t+1}) \\ (B_{t-1} + B_{t+1})/2 & (B_{t-1} \neq B_{t+1}) \end{cases} \quad (14)$$

2.4 개선된 3*3 매트릭스 알고리즘

기존의 3*3매트릭스를 이용한 알고리즘과 비슷하지만 그림 7과 같이 3번째 열의 픽셀들은 다음 매트릭스의 1번째 열의 픽셀이 되어 계산이 된다. 또한, 세로로 이동할 때도 비슷하다. 그래서 4개의 유효한 픽셀 데이터를 구하게 된다. 수식 중 R의 경우를 보인다.

	G		G	p1	p2	p3
		G		p4	p5	p6
G		G		G	p7	p8

(G의 경우)

		R		R	p1	p2	p3
					p4	p5	p6
R		R		R	p7	p8	p9

(R의 경우)

그림 7. 개선된 3*3 매트릭스 알고리즘

$$R_{12}=(p1+p3)/2 \quad (15)$$

$$R_{21}=(p1+p7)/2 \quad (16)$$

$$R_{22}=(p1+p3+p7+p9)/4 \quad (17)$$

2.5 Color Interpolation 알고리즘의 구현 및 분석

2*2 매트릭스를 이용한 알고리즘의 최대 장점은 수식이 간단하여 하드웨어로 구현이 쉽고, 메모리를 적게 사용한다. 하지만, 색상을 구현함에 있어서 4개의 이웃하는 입력 데이터에 R과 B의 유효 값이 한 픽셀뿐이라서 이들의 색상을 표현하는 것에 한계를 지닌다. 그래서 좋은 영상을 얻기 위해서는 R, B를 다시 한번 조정해주는 과정이 포함되어야 한다. 3*3 매트릭스를 이용한 알고리즘은 위의 단점이었던 R과 B의 색상표현을 일부 개선하였다. 하지만 그림 4의 (a)블록들이 4개의 유효 값을 포함한 것에 비해서 그림 4의 (b) 블록들은 2개의 유효 값을 포함하여 색상을 표현하기에 한계를 여전히 지니고 있다. 3*6 매트릭스를 이용한 알고리즘은 앞서 설명한 두 방법의 단점을 보완하였다. 즉, 18개의 이웃하는 데이터를 이용하여 R과 B의 색상 표현에 결함을 최대한으로 줄였으며 연산 실행 속도를 개선하였다. 그러나 작업을 실행할 때 계산량이 많고, 하드웨어로 구현 시 메모리를 많이 차지하는 단점을 지니고 있다. 위의 단점들을 개선한 알고리즘이 개선된 2*2매트릭스와 개선된 3*3매트릭스이다. 개선된 3*3매트릭스 알고리즘은 앞서 그림 7에서 보았듯이 R과 B의 색상을 최대한으로 표현하기 위해서 각각의 매트릭스들의 시작 포인트를 다르게 하였고, 픽셀 계산부의 단순화를 위해 2픽셀 간격으로 매트릭스가 이동한다. 이로 인해서 블록에 포함되는 유효 픽셀

값들의 수가 항상 일정하고 3*6 매트릭스를 이용한 것과의 유사한 결과를 얻을 수 있다. 또한 단순한 계산부으로 인해 하드웨어로 구현하기 쉽고 메모리도 3*6 매트릭스를 이용한 것보다 적게 차지한다. 하지만 구현을 위해서는 처리 후 계산된 값을 저장하는 라인버퍼가 별도로 필요하다. 개선된 2*2 매트릭스 알고리즘은 메모리는 기존의 2*2 매트릭스와 동일하지만 1*3 필터를 이용함으로써 R과 B가 개선되는 효과가 있다.

2.5 VHDL 구현

하드웨어로 구현 할 때 메모리의 크기, 연산수행 속도 등을 고려해야 한다. 표 1을 보면 블록 당 계산 량은 2*2 매트릭스, 화질을 보면 3*6과 개선된 3*3 매트릭스의 성능이 우수하다.

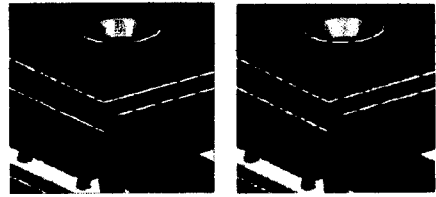
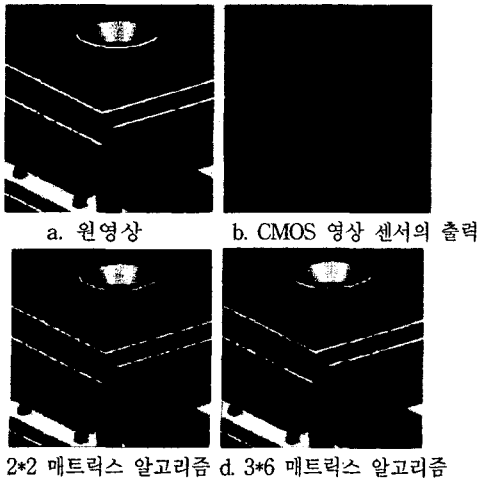
표.1 여러 알고리즘의 비교

	알고리즘				
	2*2	3*3	3*6	개선된2*2	개선된3*3
프레임계산량	50688	410112	259072	152064	152064
메모리	1	2	8	1	11
화질	하	중	상	중	상

- * 한 프레임은 352*288, 메모리는 라인버퍼단위 임
- * 계산량은 덧셈연산을 기준

그리고 그림 8을 보면 모든 알고리즘이 불완전한 영상 데이터를 입력받아 원영상과 비슷하게 복원하는 것을 볼 수 있다. 그러나 경계부분에서는 색상의 깨짐 현상이 나타나서 영상의 차이를 확실하게 알 수 있다. 3*6 매트릭스와 개선된 3*3 매트릭스가 적게 있고, 2*2 매트릭스는 색상의 깨짐 현상이 타 알고리즘에 비해 많음을 확인할 수 있다. 3*6 매트릭스는 연산부가 복잡하고 연산과정에서 사용되는 메모리가 많이 필요하다. 그러나 개선된 3*3 매트릭스는 연산부가 간단하고 메모리 할당량도 적어서 하드웨어로 구현하기 쉽다.

그림 9는 2*2 매트릭스 알고리즘을 VHDL로 설계한 후, 시뮬레이션한 결과이다. 사용된 IC는 EP1K10TC100-1이고, 로직수는 75개이다.



e. 개선된 2*2 매트릭스 알고리즘 f. 개선된 3*3 매트릭스 알고리즘
그림 8. Color Interpolation의 테스트 결과

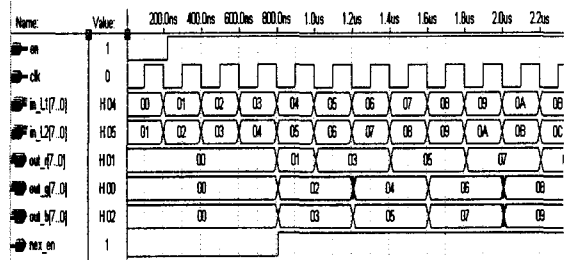


그림 9. 2*2 알고리즘 VHDL 시뮬레이션 결과

III. 결론

본 논문에서는 CMOS 영상 센서의 전처리를 위한 ISP의 Color Interpolation을 수행하는 몇 가지 알고리즘에 대해서 다루었다. 이 과정은 ISP 과정 중 픽셀의 값을 표현하는 각각의 색상을 결정하는 것으로 출력영상의 기를을 마련한다. 따라서 색상을 표현함에 있어서 충분한 데이터를 참고해야 되지만, 하드웨어의 설계를 기반으로 하기 때문에 구현의 용이성을 위해 계산식의 단순화, 사용되는 메모리의 크기, 연산수행 속도 등이 고려된다.

제시한 몇 가지의 알고리즘 중에서 구현이 용이하고 메모리를 적게 사용하는 것은 개선된 2*2 매트릭스이다. 영상의 질과 하드웨어설계를 모두 고려할 경우에는 개선된 3*3 매트릭스가 성능이 좋다. 따라서 ISP의 하드웨어를 설계하는 데에는 저가형이나 화질의 질을 크게 고려하지 않는 경우에는 개선된 2*2 매트릭스를 이용하는 것이 바람직하고, 좋은 화질을 원하는 경우에는 개선된 3*3 매트릭스를 이용한 알고리즘을 적용시키는 것이 가장 올바른 판단이라고 제안한다.

IV. 참고문헌

- [1] New interpolation method using discriminated color correlation for digital still cameras- Kuno T, Sugiura H, Matoba N
- [2] Some results in linear interpolation theory, -S.Kay, IEEE Trans. Acoust., Speech, Signal Processing, vol. 31, pp. 746-749, June 1983.
- [3] Interpolation methods for time-delay estimation using cross-correlation method for blood velocity measurement -Lai XM, Torp H