

# CMOS 이미지 센서용 효과적인 이미지 스케일 구현

이동훈, 손승일

한신대학교

## Implementation of an Efficient Image Scaler for CMOS Image Sensor

Dong-Hun Lee · Seung Il Sonh

Hanshin University

E-mail : asickorea@hs.ac.kr

### 요 약

본 논문은 CMOS 센서의 ISP 전처리 과정 후 최종 화면에 출력하기 위한 효과적인 이미지 스케일 블록을 저전력, 저비용에 맞춘 독립된 하드웨어 장치로 설계 하고자 한다. 카메라 센서 이미지 결과를 디스플레이 장치(OSD(On Screen Display))에 맞는 화면의 크기는 CIF(352x288), QCIF(176x144) 출력 모드를 사용한다. 최근 DMB 휴대용 멀티미디어 데이터 전송 사이즈 포맷에서도 위와 같은 사이즈를 지원하고 있다. 일반적인 스케일 처리에서는 PC 그래픽 카드(Graphic Card)장치의 지원을 받아서 처리하는 경우가 많다. 또는 CPU의 연산을 통한 CPU 자원을 점유하여 이미지 스케일을 처리하였다. 휴대용 CMOS 센서용에 적합한 독립적으로 처리할 수 있는 이미지 스케일 기능을 하드웨어로 설계하여 효과적인 시스템 운용과 고속 이미지 스케일 처리가 가능한 하드웨어를 설계하는게 목적이다. 이를 구현 하기위해 기존 알고리즘과 제안한 알고리즘을 비교하여 최적화된 알고리즘 적용하여 VHDL설계언어를 이용한 하드웨어 설계 후, ModelSim 6.0a를 이용하여 데이터를 검증한다.

### 키워드

Bilinear Interpolation, Bicubic Interpolation, Nearest Interpolation, SF(Scale Factor)

## 1. 서 론

요즘 카메라폰의 사용자는 복합된 기능과 고화질, 소형화, 저비용, 저전력등을 요구하는 추세이며, 이제 경쟁력있는 제품을 기업들은 단기간에 생산하여 소비자 욕구를 충족시키고자 한다. CMOS 이미지 센서는 칩 공정과정에서 공정비용이 저렴하며, 대량생산이 가능하고 다양한 기능블록 모듈을 하나로 통합하여 SoC (System on Chip)화된 제품으로 만들 수 있는 장점을 갖고 있다. 또한 이미지 영상을 출력하기 위한 디스플레이 장치는 현재 LCD(Liquid Crystal Display)를 가장 많이 사용한다. 기존의 CRT보다 높은 고화질, 경량화, 저전력 소비등의 우수한 특성 때문에 그 응용분야가 더욱 확대되고 있다. 이러한 LCD를 채용한 디스플레이 시스템은 응용분야에 따라 다양한 해상도와 고화질을 요구하고 있으며, 휴대용 컴퓨터와 모바일 기기 등에 장착되고 있는 실

정이다. 이러한 기능을 지원하기 위해 다기능, 고성능 처리가 가능한 이미지 처리 기술이 필요하다. 본 논문에서는 CMOS 센서의 이미지 처리 프로세싱(ISP)의 마지막 처리 기능인 이미지 스케일 기능에 대한 처리를 수행 하고자 한다. 이미지가 ISP블록을 거쳐 최종단에 이미지 데이터를 프레임 버퍼에 저장한 후에 출력장치의 다양한 포맷의 요구를 지원하며 R, G, B 포맷의 데이터 결과를 출력한다. 모바일 휴대폰 사이즈는 CIF(352x288), QCIF(176x144), 일반 PC에서 사용하는 CRT, LCD 컴퓨터 디스플레이 장치(1280\*1024, 1024\*768)의 해상도에 맞게 이미지 스케일을 처리한다. 이러한 이미지 스케일 처리를 기존의 소프트웨어의 응용 소프트웨어에서 처리가 가능하나 저비용, 저전력 장치에는 적합하지 않다. 결과적으로 하드웨어를 지원하는 그래픽 처리에서 주 메인 처리 기능인 CPU(Central Processing Unit), GPU(Graphics Processing Unit)의 자원을 활용하지 않고 자체

이미지 스케일 기능블록을 독립된 하드웨어로 구현하고자 한다. 다음은 다양한 이미지 스케일 알고리즘에 대해 알아보자. 이미지 스케일 처리방법에는 기존에 많이 사용하는 방법으로 가장 인접한 이웃 화소 보간법(Nearest Neighbor Interpolation), 양선형 보간법(Bilinear Interpolation), 3차 회선 보간법(Cubic Convolution Interpolation), B-스플라인 보간법(B-Spline Interpolation)기법을 사용하였다[1][2].

본 논문에서 제안한 알고리즘은 가장 인접한 이웃 화소 보간법의 장점인 빠른 픽셀 처리와 양선형 보간법의 이미지 잡음(Noise)을 줄일 수 있고 부드러운 이미지 처리(Blur)가 가능한 두 가지 장점을 갖는 이미지 스케일을 구현하였다. 각각의 알고리즘에 대해 처리된 영상 데이터 이미지를 분석 비교하여 각 특징에서 가장 저전력, 저비용, 고화질의 처리가 가능한 알고리즘을 VHDL 하드웨어 설계언어를 이용하여 설계하고 ModeSim6.0a를 이용한 데이터 분석하고 결론을 내리고자 한다[4].

## II. 본 론

### 2. 1 기본 개념(Basic Concept)

제안한 알고리즘의 기본 개념 설명이다. 이미지 스케일의 방법에서 제안한 알고리즘은 영역 픽셀 모델(Area Pixel model)과 도메인 필터 방법(Domain filtering method)을 사용한다. 이미지 스케일-UP/DOWN기능 처리에서 원본 이미지가 최소 하나에서 최대 4개의 영역을 포함하여 새로운 하나의 이미지 픽셀을 구한다[1][2].

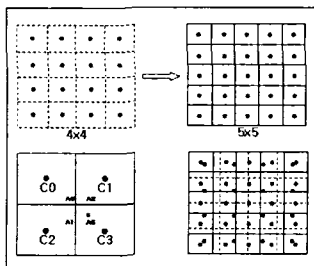


그림 1. 이미지 스케일 업(Image Scale UP)

원본 이미지 크기로 부터 새로운 이미지 크기의 한 픽셀의 필터 윈도우 요소(A Filter Windows)를 구한다. 필터 윈도우 요소는 연속적인 입력 데이터로부터 스캔 라인(Scan-Line)방향에 따라 처리한다. 그림 x1은 원본 이미지(4x4)에서 새로운 이미지(5x5)스케일 업 처리 과정이다. 원본 인접한 픽셀들은 최소 하나에서 최대 4개까지를 포함하여 새로운 픽셀 하나를 생성한다. 이유는 그림

1에 나타난것 처럼 새로운 이미지의 분할영역이 원본 이미지영역 보다 적으며 원본 이미지 영상의 픽셀이 겹쳐져 있기 때문이다. 그림 1 처럼 원본 이미지 영역들이 (C0, C1, C2, C3)가 새로운 이미지 영역에 겹쳐(Overlaped)있다. 또한 새로운 이미지 영역에 겹쳐있는 분할영역의 크기를(A0, A1, A2, A3)로 나타낸다. 가중치 계산을 위해 원본 이미지 분할영역의 크기를 1.0의 상수값으로 정한다. 그러므로 다음과 같은 수식을 구할 수 있다.

$$P = SF \times (A0 \cdot C0 + A1 \cdot C1 + A2 \cdot C2 + A3 \cdot C3) = w0 \cdot C0 + w1 \cdot C1 + w2 \cdot C2 + w3 \cdot C3 \text{-----식(1)}$$

$$SF = 1 / \text{area of filter window} \text{-----식(2)}$$

식(1)은 새로운 픽셀 하나의 구하기위한 수식이다. 식(2)는 새로운 픽셀 하나의 스케일 상수값으로 사용한다. 또한 이미지 스케일 다운(Scale-down) 처리과정도 스케일 업과정과 같다. 제안한 알고리즘의 특징은 원본 소스의 영역 픽셀 하나를 사용하였을때 가장 인접한 방법(Nearest neighbor model)과 멀티 원본 소스 영역의 픽셀 하나 이상을 사용하였을때 양선형(Bilinear model)의 두 영역 처리가 가능하여 두 가지의 장점을 갖는 알고리즘 기술이다[1][2].

### 2.2 세부 기능(Detailed Functions)

다음과 같은 세부 기능 블록으로 나누어져 있다. 디코더 스케일 요소 블록(Decoding Scaler Factor Block)은 초기값을 설정하기 위한 블록이다. SF(Scale Factor)의 초기 설정값, 새로운 이미지에 대한 필터 윈도우(A Filter Windows)의 값을 구한다. 계산식은 다음과 같다.

$$\begin{aligned} \text{winW}(\text{filter window width}) &= 1 / \text{horizontal scaleup ratio} \\ \text{winH}(\text{filter window height}) &= 1 / \text{vertical scaleup ratio} \text{-----식(3)} \\ SF = \text{winW} \times \text{winH} \text{-----식(4)} \end{aligned}$$

식(3)에서 WinW, WinH는 새로운 픽셀의 가로, 세로 크기의 계산식이며, 식(4)는 새로운 픽셀의 넓이의 계산값이다. SF상수값과 HScale ratio, VScale ratio상수값은 초기 타입에 설정된다. 라인 버퍼 블록(Line Buffer Block)은 필터 윈도우 개수 6개 좌표값을 가지며 입력 데이터 스트림(Data Stream)의 스캔-라인 처리에 따라 Horizontal, Vertical방향에서 좌표값이 증가한다. 두 개의 라인 버퍼 램을 사용하여 Horizontal, Vertical방향의 램덤으로 변화는 원본 소스 좌표값을 메모리 주소 생성블록을 통해 제어한다. 윈필터 블록(WinFilter Block)은 원본 소스로부터 다중 영역이 최대 4개 겹쳐 있을때 가중치 합을 구하기 위한 처리이다. 4개의 곱셈기와 3개의 가산기가 필요하다.

III. 제안한 알고리즘 구현

3.1 제안한 알고리즘 블록(1)

원본 이미지 크기에서 새로운 이미지 크기에 필요한 초기 변수(Horizontal ratio, Vertical ratio, Source Width, Source Height, Sacle Factor, Winw, WinH)에 대한 초기 설정 타임값을 설정한다. 앞단 블록은 초기변수에 대한 처리를 수행하기 위해 곱셈기 2개와 쉬프트 2개를 이용하였다.

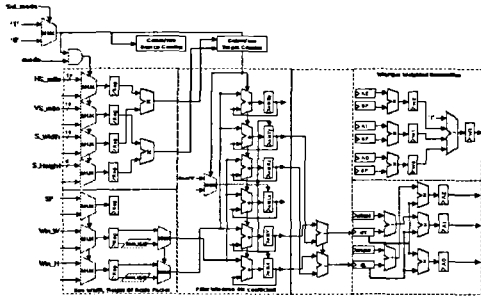


그림 2. 제안한 알고리즘 세부 블록(1)

중간 블록에서는 필터 윈도우 6개의 좌표 개수의 값이 스캔라인에 따라 증가함에 따라 6개의 가산기를 이용하였다. 마지막 블록은 새로운 필터 윈도우 크기영역에 겹쳐진 원본 소스 블록의 다중 영역의 크기를 구하기 위해 3개의 곱셈기와 4개의 감산기를 이용하고, 새로운 이미지 픽셀의 겹쳐진 가중치 합을 구하기 위해 3개의 곱셈기와 3개의 가산기를 필요로 한다. 위 그림 2는 제안한 알고리즘 상세 처리블록(1)을 나타내고 있다[3][4].

3.2 제안한 알고리즘 블록(2)

필터 윈도우 개수 6개의 좌표값이 입력 스트림 데이터에서 스캔라인 방향으로 이동하면서 처음 연산 하기 시작하는 위치는 두 번째 라인의 입력이 끝나는 시점에서 새로운 첫 라인이 처리한다. 그러므로 첫 번째 라인 버퍼 램에 저장하고 두 번째는 두 번째 라인 버퍼 램에 저장한 후 두 개의 데이터를 동시에 읽어 처리하게 하였다. 또한

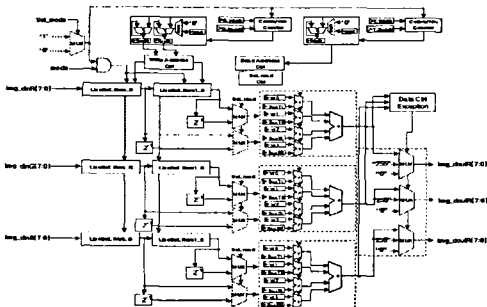


그림 3. 제안한 알고리즘 세부 블록(2)

연속 스트림 데이터를 처리하기 위해 Ping-Pong 램 액세스 방법을 사용하였다. 출력결과를 위해 픽셀 영역 가중치와 원본 이미지 픽셀값을 읽어 곱셈기 4개와 가산기 3개를 이용하여 출력 결과를 얻도록 설계하였다[3][4].

IV. 실험 및 고찰

4.1 시뮬레이션 분석 및 검증

그림 4에서 실험영상 데이터는 CIF(352x288) 원본 크기의 이미지를 입력하여 QCIF(176x144) 크기의 새로운 이미지 처리에 대한 시뮬레이션을 수행하였다. 출력 결과 타임은 첫 라인과 두 번째 라인의 데이터 메모리 쓰기 액세스 이후부터 어드레스 발생기의 컨트롤(Address Generator Control)이 가능한 시점 이후부터 결과를 출력한다.

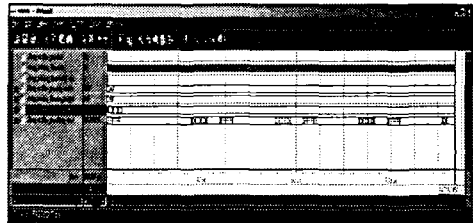


그림 4. 시뮬레이션 파형 검증

제안한 알고리즘은 하드웨어 설계 언어인 VHDL를 사용하였고, XCV1000e-6hp240계열에서 최대 63.155Mhz클럭의 속도로 처리 되었다.

아래 표 1은 제안한 알고리즘에서 사용한 컴포넌트 사용수를 나타내고 있다.

표 1. 연산에 필요한 컴포넌트 수의 비교

	Nearest Neighbor	Bilinear	제안한 알고리즘	Bicubic
픽셀 이동좌표	2Add	2Add	6Add	2Add
가중치 요소	0Mult 0Add	3Mult 2Add	6Mult 2Add	16Mult 36Add
가중치의 합 계산	0Mult 0Add	4Mult 3Add	4Mult 3Add	16Mult 15Add
메모리 액세스	1Read 1Write	4Read 1Write	4Read 1Write	16Read 1Write

4.2 데이터 분석

다음은 영상의 이미지를 부분 확대하여 영상의 일그러짐 정도와 영상의 부드러운 이미지의 차이

를 비교 하였다. 이미지는 Lena 표준영상을 사용 하였다. 그림 x의 (a)는 Lena 원본영상 256x256의 크기에서 Lena의 특징적인 눈을 확대하여 다양한 스케일 알고리즘으로 처리한 데이터들을 비교하였다. 그림 (b)는 가장 이웃한 화소 픽셀 알고리즘(Nearest Interpolation)을 적용하였다. 이웃한 픽셀값을 바로 복사하여 사용하기 때문에 확대한 이미지에서 경계선 처리 문제에 대한 일그러짐 현상이 보였고 에지는 전체적으로 잘 나타났다. 그림 (c)는 3차 회선 보간법(Bicubic Interpolation)을 적용하여 비교 대상 알고리즘에서 가장 좋은 영상의 화질과 에지를 나타내었다. 그림(d)에서는 양선형 보간법(Bilinear Interpolation)을 사용하였다. 양선형 보간법에서는 영상의 확대하여 보았을 때 부드러운 이미지 영상으로 나타나지만 영상의 에지(Edge) 검출한 이미지에 대해서는 에지 처리 가장 잘 나타나지 않았다. 그림(e)에서는 제안한 알고리즘을 사용 하였다. 양선형 보간법의 부드러운 이미지 영상과 가장 인접한 픽셀 보간법의 에지 특징을 잘 나타내어져 두 가지의 장점을 가진 알고리즘으로 양선형(Bilinear)과 비교하여 연산량이 비슷하지만 영상의 이미지는 더 좋은 화질의 결과를 보였다[2][3].



(a) 원본



(b) Nearest



(c) Bicubic



(d) Bilinear



(e) 제안한 알고리즘

그림 5. 이미지 스케일 업과 에지 검출 비교

## V. 결 론

본 논문에서는 CMOS 센서용에 적합한 이미지 스케일러를 설계 하였다. 기존의 다양한 이미지 스케일러 알고리즘 처리에서 더 좋은 화질 개선을 요구하면 칩의 비용이 비싸지고 그 반대일 경우는 좋은 화질을 얻을 수 없다. 제안한 알고리즘은 기존의 양선형 스케일(Bilinear Scaler)과 비슷한 비용에서 보다 좋은 화질로 개선할 수 있게 처리하였다. 또한 표 1에서 보는것과 같이 비슷한 연산량을 필요하면서 양선형과 가장 인접한 픽셀

처리 두가지 장점을 가지며 저전력, 저비용, 고품질 측면에서 효과적인 이미지 스케일 기능을 수행한다. 결과적으로 본논문에서 제공하는 알고리즘 구현은 CMOS 센서용, 모바일 디스플레이 장치의 저전력, 저비용에 적합한 이미지 스케일 응용 모델로 사용될 수 있다고 사료된다[1][5].

## 참고문헌

- [1]. Chun-Ho Kim, Si-Mun Seong, Jin-Aeon Lee, and Lee-Sup Kim, Winscale : An Image-Scaling Algorithm Using an Area Pixel Model, IEEE Trans. Circuits and Systems For Video Techninology, VOL. 13, June 2003.
- [2]. Randy Crane, Hewlett-Packard Company. "A Simplified Approach to Image Processing". Prentice Hall PTR.
- [3]. Rafael C. Gonzalez, Richard E. Woods. "Digital Image Processing". Addison Wesley.
- [4]. 조태경, 홍재인 AMBA기반의 LCD컨트롤러 설계 한국콘텐츠학회 논문지 '04 Vol. 4 No 4.
- [5]. S. Ramachandran S.Srinivasan "Design and FPGA Implementation of a Video Scaler with on-chip reduced memory utilization". DSD'03. IEEE 2003.