

論文2003-40SD-3-5

면적 점유비를 이용한 영상 스케일러의 설계

(A Hardware Implementation of Image Scaler Based on Area Coverage Ratio)

成始門*, 李珍彦*, 金春昊*, 金利變*

(Si-Mun Seong, Jin-Aeon Lee, Chun-Ho Kim, and Lee-Sup Kim)

요약

TFT LCD 와 같은 디지털 디스플레이 디바이스는 CRT 와 같은 아날로그 디스플레이 디바이스와 달리 그 제조 과정에서부터 해상도가 정해져 버리는 단점을 가지게 된다. 그러나 이들 디스플레이 디바이스에 출력이 되는 입력 화면의 해상도의 종류는 매우 다양하며 출력 디바이스의 해상도 또한 날로 다양해지고 있다. 이러한 입력 영상의 해상도를 출력 영상의 해상도에 맞게 스케일을 늘리거나 줄이는 일(interpolation / decimation)을 하는 것을 영상 스케일러라고 한다. 이러한 스케일 up/down 과정에서 생길 수 있는 영상의 열화를 줄이기 위한 알고리즘과 이를 이용한 H/W cost 가 저렴한 영상 스케일러에 대한 연구가 기존에 진행되어 왔다. 본 논문에서는 영상 scale up/down에 있어서 이상적이라 할 수 있는 연속 공간에서의 광학적 영상 확대/축소를 이산 공간인 디지털 디스플레이 디바이스에 맞게 옴진 Winscale 알고리즘을 제안한다. 그리고 제안된 알고리즘을 이용한 영상 스케일러를 Verilog XL 을 이용해서 H/W로 구현하였다. 그리고 삼성 SOG 0.5um 공정을 이용하여 실제 칩으로 제작되었다. 기존의 다른 소프트웨어에서 사용되고 있는 영상 스케일링 알고리즘을 이용해서 스케일된 영상의 R, G, B 각 칼라 채널에 대한 PSNR 값을 가지고 스케일링 기능의 우열을 비교했다. 또한 H/W cost 도 비교하였다. 이러한 Winscale 방법을 이용한 영상 스케일링은 영상 품질은 기존의 알고리즘과 비등하거나 우수하면서 H/W cost 가 기존의 것들 보다 저렴하기 때문에 영상 스케일러가 필요한 다양한 디지털 디스플레이 디바이스에 사용될 수 있을 것이다.

Abstract

Unlike in analog display devices, the physical screen resolution in digital devices are fixed from the manufacturing. It is a weak point on digital devices. The screen resolution displayed in digital display devices is varied. Thus, interpolation or decimation of the resolution on the display is needed to make the input pixels equal to the screen resolution. This process is called image scaling. Many researches have been developed to reduce the hardware cost and distortion of the image of image scaling algorithm. In this paper, we proposed a Winscale algorithm, which modifies the scale up/down in continuous domain to the scale up/down in discrete domain. Thus, the algorithm is suitable to digital display devices. Hardware implementation of the image scaler is performed using Verilog XL and chip is fabricated in a 0.5 um Samsung SOG technology. The hardware costs as well as the scalabilities are compared with the conventional image scaling algorithms that are used in other software. This Winscale algorithm is proved more scalable than other image-scaling algorithm, which has similar H/W cost. This image-scaling algorithm can be used in various digital display devices that need image scaling process.

Keywords : winscale , image scaler , hardware implementation

* 正會員, 韓國科學技術院 電子電算學科 電氣및電子工學
(Division of Electrical Engineering, Dept. of Electrical
Engineering & Computer Science, KAIST)
接受日: 2001年3月2日, 수정완료일: 2003年2月28日

I. 서론

최근 이동형 디바이스의 공급과 노트북 컴퓨터가 전체 컴퓨터의 20%를 넘을 정도로 증가하게 되면서 flat

panel monitor 의 급격한 증가가 이루어 지고 있으며 가격 또한 날로 떨어지고 있는 추세이다. 요즘 사용되고 있는 FPM(flat panel monitor)은 그래픽스 프레임버퍼의 디지털 정보를 직접 FPM의 column 구동기와 row 구동기에 디지털로 인터페이스한다. 그러나 기존의 FPM들은 디스플레이 되기 전에 프레임버퍼에 들어있는 디지털 영상 정보를 D/A 컨버터를 이용해서 우선 써리얼의 아날로그 스트림으로 바뀌어서 케이블로 전송하고 이를 다시 A/D 컨버터를 이용해서 디지털 영상을 복원해서 디스플레이 하게 된다. 이러한 변환과정과 전송과정에서 영상의 품질이 떨어질 수 있으며 정보를 잃을 수도 있다. 그러나 디지털 신호는 그 자체로 노이즈를 견디지 않는다. 이러한 아날로그 인터페이스의 단점으로 인해서 서서히 CRT를 FPM의 대표인 LCD가 대신하고 있다.

아날로그 신호로 좋은 화질의 영상을 효과적으로 볼 수 있는 해상도는 XGA(1024×768)에서 SXGA (1280×1024)가 한계라고 한다. 이에 LCD의 해상도가 1600×1200으로 바뀌는 급변 정도에는 상당수의 CRT를 LCD가 대체할 것으로 보인다.

LCD와 같은 디지털 디스플레이 디바이스의 단점은 그 생산 단계에서 CRT와는 달리 해상도가 정해져 버린다는 것이다. 디지털 디바이스에는 <표 1>과 같은 resolution naming scheme이 있으며, 디지털 디바이스의 해상도와 이에 입력으로 들어오게 되는 영상의 해상도 또한 <표 1>의 해상도 중 하나가 되게 된다.

표 1. 해상도 명명 방법
Table 1. Resolution Naming Scheme.

Resolution Name	Horizontal Resolution X Vertical Resolution
VGA	640 X 480
XGA	1024 X 768
SXGA	1280 X 1024
UXGA	1600 X 1200
HDTV	1920 X 1080
HDTVplus	1920 X 1200
QXGA	2048 X 1536

이러한 정해진 디지털 디스플레이 디바이스의 입력으로는 다양한 해상도의 입력 영상이 들어오게 되며 또한 디지털 디스플레이 디바이스의 해상도 또한 날로 다양해지고 있다. 이러한 해상도의 차이를 인터페이스(확대 혹은 축소)하는 것을 영상 스케일러라고 한다.

이러한 영상 스케일러가 갖추어야 되는 조건은 영상의 확대 혹은 축소과정에서 열화가 일어나서는 안되며, H/W가 커서는 안 된다는 것이다. 또한 vertical 스케일링 과정에서 불가피하게 필요하게 되는 라인버퍼의 크기 또한 커서는 안 된다.

본 논문에서는 영상 확대 혹은 축소에 있어서 연속 영역(continuous domain)에서의 영상 확대/축소를 이산 영역(discrete domain)인 디지털 디스플레이 디바이스에 맞게 옮긴 Winscale이란 새로운 알고리즘을 제안한다. 그리고 제안된 알고리즘을 이용한 영상 스케일러를 Verilog XL을 이용해서 H/W로 구현하였다. 그리고 PCI 인터페이스 버전의 winscale 영상 스케일러의 코어는 삼성 SOG 0.5um 공정을 이용하여 실제 칩으로 제작되었다.

II. Winscale 알고리즘의 영상 스케일러의 설계

1. Winscale 알고리즘

Winscale 알고리즘은 기본적으로 연속 영역 필터링(continuous domain filtering) 및 보간(interpolation) 기법을 이용해서 영상을 확대 혹은 축소를 한다. <그림 1>은 간략히 4×4 해상도의 영상을 5×5 해상도의 영상으로 확대하는 예를 나타내는 개념도이다.

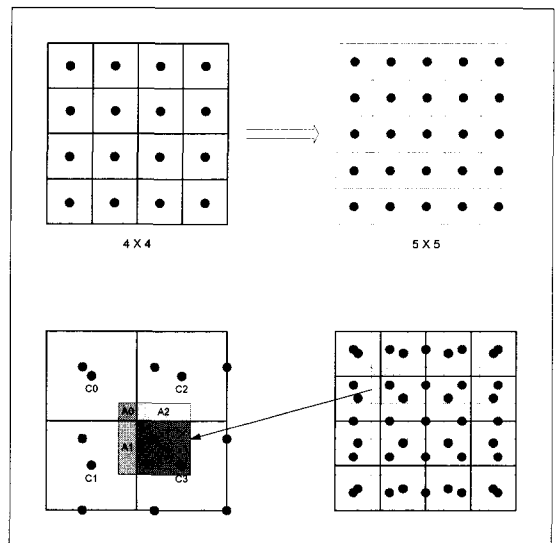


그림 1. 영상확대의 개념도
Fig. 1. concept of image scaling.

<그림 1>의 좌측 하단의 그림은 확대하여 얻는 출력 영상의 화소값은 필터윈도우에 걸치는 주위의 입력 영상의 각 화소값을 필터 윈도우가 점유하는 화소값의 면적비를 가중치로 써서 구한다는 것을 보이고 있다.

즉 결과 영상의 화소(왼쪽 하단 그림의 가운데 화소(A3 부분)를 두르고 있는 윈도우가 입력 영상의 화소인 C0~C3를 두르고 있는 윈도우를 점유하는 면적을 구하고 이 점유하는 면적의 비에 따라서 각 C0~C3의 화소 값을 결과 영상의 화소 값에 반영하는 것이다.

<그림 2>는 5×5의 해상도를 가지는 영상을 4×4의 해상도를 가지는 영상으로 축소하는 예를 보여주고 있는 개념도이다. 축소가 확대의 경우와 다른 점은 필터윈도우의 크기가 가로 세로 각각 1.0으로 고정되어 있고, 필터윈도우의 이동이 1.0 보다 크다는 것이다. 축소도 확대와 마찬가지로 수평 또는 수직 방향으로 각각 독립적으로 축소할 수 있으며 축소비가 1/2 보다 더 크게 축소를 해야 하는 경우는 prescaler에서 미리 2의 지수승 만큼 축소하여 축소비가 1/1에서 1/2 사이에 놓이도록 한다. 이와 같이 하면 다음 예와 같이 축소 필터윈도우에 한번이라도 걸리지 않는 화소가 존재하지 않게 된다.

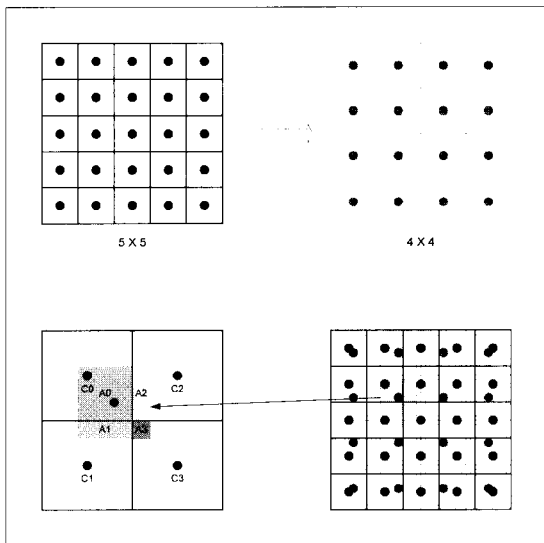


그림 2. 영상축소의 개념도
Fig. 2. concept of image reducing.

<그림 3>은 영상 스케일러의 전체적인 동작을 블록 다이어그램으로 보인 것이다. 제안하는 영상 스케일러

는 크게 네 부분으로 구성되어 있다. 영상 필터의 계수를 생성하는 부분 (필터윈도우보간기 및 필터계수발생기), 영상의 축소 배율이 클 때만 사용되는 부분인 prescaler, 그리고 수직 스케일링을 위해 라인을 임시로 저장하는 부분인 라인버퍼, 그리고 최종 스케일링된 영상을 생성하는 부분인 필터(weighted sum 수행) 부분으로 구성된다.

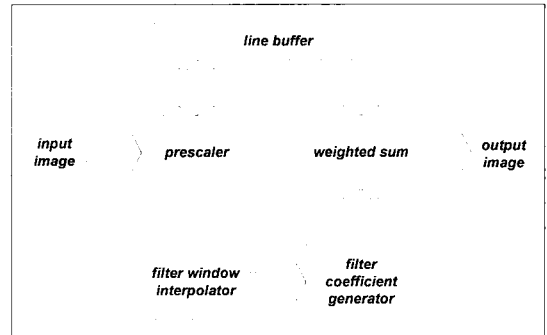


그림 3. 영상 스케일러의 전체적인 동작 flow
Fig. 3. overall data flow of image scaler.

필터 계수의 계산과정을 <그림 4>를 보면서 좀더 상세히 살펴보자.

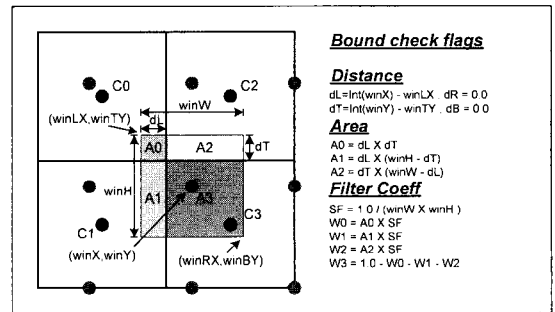


그림 4. 영상 스케일러 면적 점유비 및 필터계수의 계산
Fig. 4. calculation area coverage ratio & filter coefficients.

<그림 4>의 필터윈도우의 위치(winX,winY)는 <그림 3>의 필터윈도우보간기의 덧셈기에 의해서 계산되고, 필터계수발생기에서 필터윈도우의 점유면적 (A0~A3)을 구한 다음에 이 면적 값에 스케일팩터(SF, 확대/축소 비율)를 곱하여 필터 계수를 구한다. 실제 하드웨어 구현에서는 필터 계수로 8-bit 값을 사용하기 때문에 면적 값도 8-bit 정도의 precision 을 가지면 된다.

한 영상 화소의 값을 얻기 위해서는 필터링에 포함 되는 원래 영상의 화소는 4 개가 최대이며 이를 수직 방향으로 보면 두 줄의 영상 화소의 라인이 필요한 것을 알 수 있다. 즉, 영상 화소의 확대 혹은 축소 배율에 관계 없이 단 두 줄의 라인버퍼만 있으면 영상 스케일링이 가능함을 알 수 있다.

필터 윈도우의 이동은 선형 보간 기법(linear interpolation method)을 이용하여 간단히 해결할 수 있다. 즉, 필터 윈도우의 이동은 수평 방향일 때는 필터 윈도우의 x 좌표 값에 필터 윈도우의 너비만큼을 더하고, 수직 이동일 때는 y 좌표 값에 필터 윈도우의 높이를 더하면 된다. 이때 필터링 계수를 면적 점유비에 서 구하기 때문에 영상 화소의 부분 점유면적(A0~A3) 계산을 용이하게 하도록 필터의 중심과 네 꼭지점의 좌표를 모두 필터윈도우보간기에 설정하여 수평 또는 수직 방향 이동에 따라 갱신되도록 한다. 부분 점유 면적 계산은 필터윈도우보간기의 출력 값에서 계산할 수 있다. 부분면적은 필터 윈도우의 네 꼭지 점의 정수 값 과 필터 윈도우 중심좌표의 정수 값을 비교하여 필터 윈도우의 중심이 위치하는 영상 화소와 상하좌우로 인접한 영상 화소에 필터 윈도우가 걸치는지 검사하고 (<그림 4>의 경우 XL = True , XR = False , YT = True , YB = False , 즉 필터 윈도우 중심에서 왼쪽 및 위쪽 영상 화소에 필터 윈도우가 걸쳐져 있음) 걸치는 경우 인접하는 영상 화소 상에 필터 윈도우가 걸치는 거리 값 (<그림 4>의 dL 과 dT)을 계산하여 둔다. 그 다음에 이 거리 값을 이용하여 인접한 영상 화소에 걸치는 필터 윈도우의 부분 면적 값을 구한다. 이렇게 구한 면적의 비율을 필터 계수로 써서 다음과 같이 최종 필터링 된 영상을 얻는다(<그림 3>의 weighted sum block).

$$C=C0 \times W0+C1W1+C2W2+C3 \times W3$$

이 때 필터 계수는 면적비를 그대로 사용하여 필터링 할 수도 있고, 더 넓은 면적을 차지하는 영상 화소에 더욱 가중치를 주는 방법이 가능하다. 후자의 경우는 면적 점유비를 큰 순서대로 정렬한 다음 면적비가 작은 쪽의 계수 값을 일정량 빼서 큰 면적비를 갖는 계수에 더하는 방법으로 가중치를 변경하여 적용할 수 있다.

필터 윈도우의 크기는 x , y 각 방향의 확대 배율에

따라 결정된다. 원래 영상의 한 영상 화소의 높이와 너비를 각각 1로 하였을 때 필터 윈도우의 너비와 높이는 다음과 같다.

$$\text{필터 윈도우의 너비} = 1.0 / (\text{X 방향 확대 배율})$$

$$\text{필터 윈도우의 높이} = 1.0 / (\text{Y 방향 확대 배율})$$

<그림 2>는 영상을 축소할 때의 개념도를 보인 것이다. prescaler를 제외한 전체적인 구성은 영상을 확대하는 경우와 같다. 이 경우 필터 윈도우의 크기는 너비와 높이를 각각 1.0으로 설정하여, 필터윈도우보간기의 증분(incremental value)은 1 보다 큰 값으로 설정된다. Prescaler 의 축소비율과 필터윈도우보간기의 증분은 x, y 방향에서 서로 독립적으로 계산하며 다음과 같이 구한다.

$$WRatio = \text{입력영상 수평해상도} / \text{출력영상 수평해상도}$$

$$\text{정수 } w : 2^w \leq WRatio < 2^{w+1}$$

$$\text{prescaler 수평방향 정수배 scale factor} = 2^w$$

$$\text{필터 윈도우 } \times \text{ 방향 증분} = \text{입력영상 수평해상도} / (\text{출력영상 수평해상도} \times 2^w)$$

$$hRatio = \text{입력영상 수직해상도} / \text{출력영상 수직해상도}$$

$$\text{정수 } h : 2^h \leq hRatio < 2^{h+1}$$

$$\text{prescaler 수직방향 정수배 scale factor} = 2^h$$

$$\text{필터 윈도우 } y \text{ 방향 증분} = \text{입력영상 수직해상도} / (\text{출력영상 수직해상도} \times 2^h)$$

prescaler는 하나의 라인 버퍼와 입력되는 영상 신호 값을 x, y 방향의 정수배 축소비율 만큼 더하기 위한 덧셈기로 구성할 수 있다.

2. H/W 구현

Winscaler 알고리즘을 이용한 영상 스케일러는 CADENCE 의 verilog XL 을 이용해서 구현되었으며, synopsys tool을 이용해 합성되었다. PCI 버스를 이용해서 host PC와의 인터페이스 하는 winscale 알고리즘의 영상 스케일러는 <그림 5>와 같은 구조를 가지고 있다. 앞에서 보인 <그림 3>의 동작 flow대로 H/W 블록들이 구성되어 있는 것을 볼 수 있다.

영상 스케일러의 앞 단에는 host PC와 PCI bus를 통

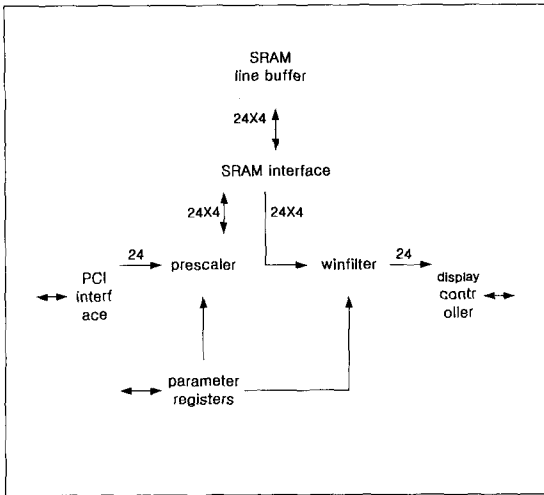


그림 5. 영상 스케일러 H/W 블록 다이어그램
Fig. 5. hardware block diagram of image scaler.

해서 인터페이스 할 수 있게 PCI 인터페이스가 붙으며, 뒤 단에는 LCD-panel의 동작 타이밍에 맞게 스케일링 된 영상을 보낼 수 있게 디스플레이 제어가 붙는다. 각각의 인터페이스와 winscale 영상 스케일러의 core 간에는 FIFO를 박고 2 개의 FIFO에서 나오는 신호를 영상 스케일러가 보고 stall 동작을 하며, 앞 단 FIFO 에는 영상 스케일러가 read_pixel 이란 신호를 보내서 host로부터 오는 화소의 값을 읽게 된다.

영상 스케일러의 core 는 삼성 0.5um 공정 KG80 라이브러리로 게이트수가 29000 이며, vertical 필터링을 위해 필요한 라인 버퍼는 2 라인이다. 보통 FIR 필터가 들어가는 스케일러의 경우 필터 tap의 길이 만큼의 라인 버퍼가 보통 들어가게 되는데 이에 비하면 굉장히 저렴한 라인 버퍼의 수이다.

prescaler와 winfilter의 좀더 상세한 H/W 블록 다이

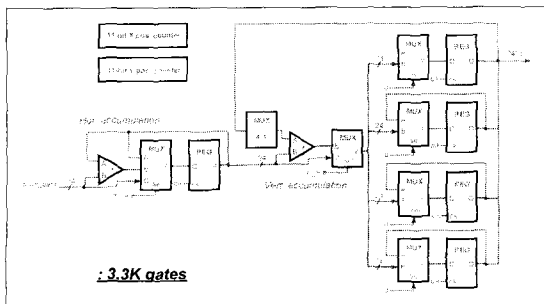


그림 6. prescaler H/W 블록 다이어그램
Fig. 6. hardware block diagram of prescaler.

이그램을 보자.

prescaler 는 2 배 이상의 축소가 필요할 경우 2 의 지수승으로 미리 축소를 수행하는 부분이며 크게 horizontal accumulation 을 수행하는 부분과 vertical accumulation 을 수행하는 부분으로 나뉘어진다. 그런데 보통 디지털 디스플레이 디바이스에 디스플레이를 하는 경우 확대를 주로 하며, 특히 2 배 이상의 축소는 거의 쓰이지 않는 기능이므로 scalability 차원에서 구현된 블록이라고 할 수 있다. Prescaler 블록의 게이트 수는 삼성 0.5um 공정 KG80 라이브러리를 이용해서 3.3K 이다.

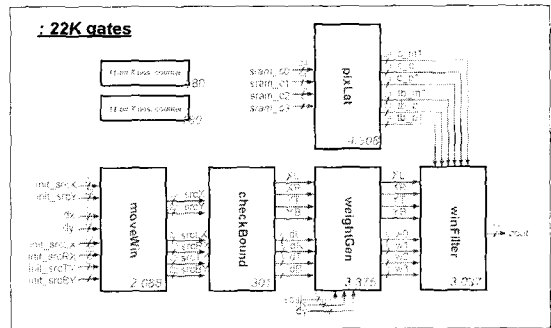


그림 7. winfilter H/W 블록 다이어그램
Fig. 7. hardware block diagram of winfilter.

Winfilter는 크게 moveWin, checkBound, weightGen, pixLat, winFilter 의 5 개의 블록으로 구성된다. 각각의 블록의 기능을 간략히 살펴보자.

우선 moveWin 은 전단의 parameter register block 에서 받은 필터의 초기 파라미터(int_srcX, int_srcY, dx, dy, int_srcLX, int_srcRX, int_srcTY, int_srcBY) 를 가지고, horizontal 및 vertical counter를 보고 필터 윈도우의 각 중심점(srcX, srcY)과 4 개의 모서리의 위치값(srcLX, srcRX, srcTY, srcBY)을 구하는 일을 수행한다.

checkBound block은 moveWin 블록으로 받은 필터 윈도우에 대한 위치 정보를 가지고 boundary check를 수행하고 flag(XL, XR, YT, YB)를 생성해서 다음 블록인 weightGen block에 전달하고 필터 윈도우 면적 점유율을 계산하는데 필요한 파라미터(dL, dR, dT, dB) 또한 생성해서 다음 블록에 전달한다.

다음의 weightGen block 은 앞 단에서 받은 flag 와 점유율 계산에 필요한 파라미터를 가지고 면적 점유율

을 계산해서 말단의 winFilter 블록에 앞 단에서 받은 flag 와 함께 전달한다.

마지막으로 pixLat 블록은 라인 버퍼에서 받은 두 줄의 필터 윈도우가 걸치게 되는 입력 영상 화소에 해당하는 6 개의 화소값을 winfilter에 보내는 블록이다. 6 개의 블록 중 4 개만 실제 필터링에 이용된다. 이 과정에서 checkBound에서 얻은 flag가 쓰인다. 이 블록의 용도는 실제 FPGA 테스트 보드 상에는 라인 버퍼를 비동기식 SRAM으로 구현했는데 다른 알고리즘의 검증을 위해서 4 줄의 라인 버퍼가 구현되어 있다. 이를 모두 사용하기 위해서 들어가는 블록이다.

실제 winscale 알고리즘을 이용한 영상 스케일러의 경우는 2 줄의 라인 버퍼만 있어도 가능하므로 , 2 줄의 라인 버퍼 이용 시에 pixLat 블록은 조정되어야 된다. Winfilter 블록의 게이트수는 삼성 0.5um 공정 KG80 라이브러리를 이용해서 22K 이다.

ASIC 화에 앞서서 FPGA를 이용한 보드 테스트를 수행했다. <그림 8>과 같은 구성도로 테스트 환경을 구성했다.

사용했다.

그러나 테스트 보드의 PCI 버스 인터페이스가 다른 디바이스와의 간헐적인 충돌을 일으켜 완벽한 640×480 영상을 볼 수는 없었으며 , PCI 대역폭 또한 원하는 만큼 나와주지 않아서 완벽한 prototyping이 되지 못했다. winscale 영상 스케일러 core는 삼성 0.5um SOG 공정을 통해서 ASIC 화가 되었다.

III. 화질 및 H/W cost 비교

1. 화질 비교

본 winscale 알고리즘은 기존의 영상 스케일링 알고리즘 중 bilinear 알고리즘과 H/W cost에 있어서 매우 비슷하다. 그래서 bilinear 영상 스케일링 알고리즘을 H/W 나 S/W 적으로 주된 비교 대상으로 한다.

영상 스케일링 알고리즘에 있어서 복잡하게는 적응적(adaptive) 알고리즘들과 단순하게는 영상 화소 값의 단순반복을 사용하는 알고리즘 등 그 종류가 매우 다양하지만, 영상 화소의 단순 반복을 이용하는 방법들은 H/W cost 가 저렴함에 비해 영상의 blocking 효과로 인해서 좋은 화질을 기대 할 수 없으며 , 복잡한 알고리즘의 경우에는 그 H/W cost 가 엄청나기 때문에 구현에 어려움이 존재하게 된다.

알고리즘의 비교는 XGA 급의 영상을 각각의 영상 스케일링 알고리즘을 이용해서 VGA로 스케일 down한 다음 이를 다시 XGA 급의 영상으로 스케일 up해서 원 영상과 비교를 하였다. 이러한 비교 방법을 선택한 이유는 우선 VGA 나 XGA 같은 크기가 큰 영상에 대해서는 각각의 해상도에 대한 원 영상이 없기 때문이다. 기존의 영상처리에서는 256×256 이나 512×512에 해당하는 흑백의 원 영상들이 몇몇 있지만 영상 스케일러 자체의 용도가 디스플레이 디바이스이기 때문에 그런 작은 크기의 영상에 대한 알고리즘 비교는 무의미하며 흑백 채널인 것도 이유 중 하나이다.

정성적으로는 영상을 유관으로 비교하며 , 정량적으로는 각각의 RGB 채널의 칼라 값의 PSNR 값을 비교 한다.

비교 알고리즘으로는 포토샵이나 페인트샵에서 영상 스케일링 알고리즘으로 사용하고 있는 NN (nearest neighbor), bilinear, bicubic를 이용하되 이를 C로 구현 함에 있어서 winscale 영상 스케일러와 H/W 상에서와

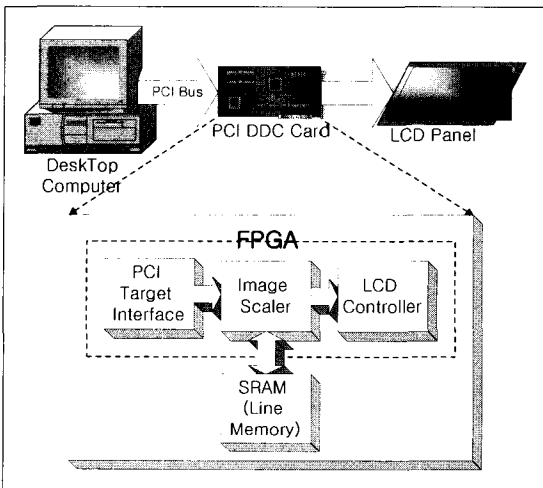


그림 8. FPGA 테스트 환경
Fig. 8. Test environment using FPGA.

<그림 8>과 같이 PCI 버스 인터페이스의 보드를 구성하고 winscale 알고리즘을 실험적으로 구현해보았다. CPLD로는 Altera Flex10k100ABC356-1, 356 pin, BGA type, speed grade-1을 사용했고 LCD panel 은 LGLP104V2-W (해상도 : 640X480) 그리고 SRAM 은 삼성 비동기식 SRAM인 K6R1016V1C 64KX16 bit를

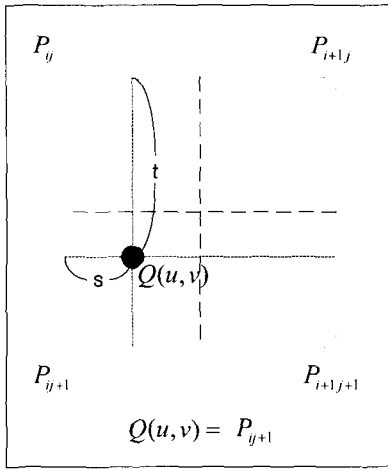


그림 9. NN (nearest neighbor)
Fig. 9. nearest neighbor algorithm.

동일한 정밀도의 제한을 주었다. 앞의 3 가지 알고리즘을 그림과 함께 잠시 살펴보자.

NN (nearest neighbor) 는 3 가지 알고리즘들 중 가장 단순한 알고리즘으로 보간되어야 하는 화소가 $Q(u,v)$ 일 때 이 값은 입력 영상 화소 값 중에서 가장 가까운 점을 그 값으로 바로 mapping한다.

<그림 9>에서는 $Q(u,v)$ 의 값이 P_{ij+1} 이 된다. 알고리즘이 간단해서 H/W 구현이 용이하지만, blocking 효과가 나타나므로 그리 좋은 스케일링 방법은 아니다. 그러나 원 영상의 화소를 그대로 가져다가 사용하기 때문에 영상이 선명한 장점이 있다.

bilinear 스케일링 알고리즘은 H/W cost 가 저렴해서 구현이 비교적 용이하면서 적당한 화질을 내는 알고리

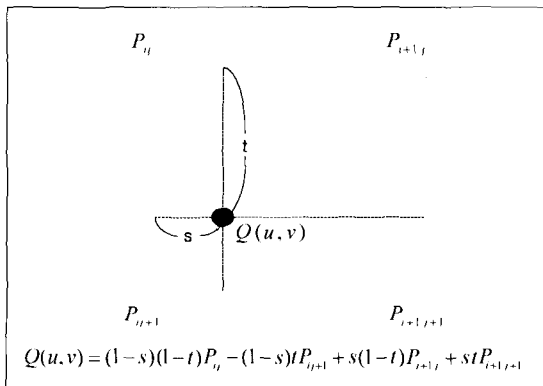


그림 10. bilinear
Fig. 10. bilinear algorithm.

즘이다. winscale 알고리즘과 비슷한 H/W cost를 보이기 때문에 bilinear 알고리즘과의 화질의 비교는 의미를 가진다.

기본적인 스케일링 방법은 출력 영상과 입력영상의 화소를 겹쳐놓고, 출력 영상 화소에서 주위의 입력 영상 화소까지의 거리를 이용해서 선형 보간(linear interpolation)을 x 와 y 방향으로 두 번 수행하는 것이다. Winscale 알고리즘과의 가장 큰 차이점은 출력 영상의 화소가 어디에 위치하든 입력 영상의 화소의 모든 값은 출력 영상의 화소에 영향을 주게 되므로 영상이 뿌옇게 흐려지는 열화 효과가 나타난다. 그러나 Winscale 알고리즘의 경우 필터윈도우의 크기에 따라서 불필요한 영향을 주지 않을 수 있다. 그러므로 좀더 선명한 영상을 얻을 수 있는 장점을 가진다.

<그림 12>와 같은 경우 bilinear 의 경우는 보간화소인 C1는 원 영상의 4 화소(P1,P2,P3,P4)의 영향을 모두 받는데 반해, winscale알고리즘의 경우 보간화소인 C1의 경우 원 영상의 P3의 영향만 받아 불필요한 blurring을 줄이게 되어 더 선명한 영상을 얻을 수 있다. 즉, bilinear알고리즘은 어떤 해상도에도 관계 없이 주위의 4 영상화소의 영향을 보간화소가 받게 되고, winscale의 경우에는 확대 비율에 따라서 보간화소 주위의 4 개의 원화소의 영향을 모두 받지 않을 수 있다.

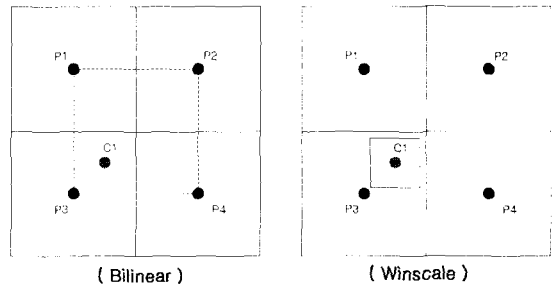


그림 11. bilinear와의 차이
Fig. 11. algorithmic difference between bilinear & winscale.

이러한 점이 bilinear 와 NN 의 조합으로 보일 수도 있다. 부연설명을 통해서 bilinear 와 winscale의 차이점을 설명하겠다. 일단 가장 일반적인 경우인 winscale에서 원화소 4 개의 점을 이용해서 보간하는 경우에 대해서 살펴보자.

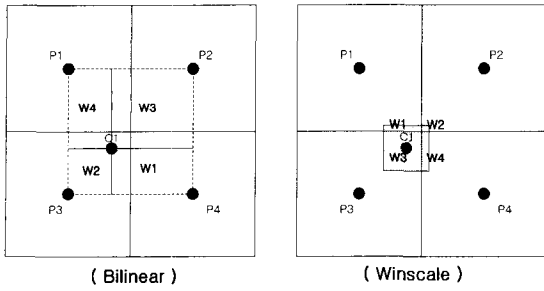


그림 12. bilinear와의 차이
Fig. 12. algorithmic difference between bilinear & winscale.

<그림 12>는 같은 배율로 확대하는 경우에서 bilinear 와 winscale의 차이점을 보일 수 있는 예제를 도식화 한 것이다. 일단 보간화소 C1 을 P1, P2, P3, P4를 이용해서 구하는 공식은 다음과 같이 동일하다.

$$C1 = (P1*W1 + P2*W2 + P3*W3 + P4*W4)$$

그렇지만 <그림 12>에서 보이듯이 bilinear에서의 점유비와 winscale에서의 점유비가 다를 수 있다. 그래서 결론적으로 얻어지는 보간화소인 C1의 값이 다르게 된다.

또한 <그림 11>의 그림과 같은 경우가 특별히 NN (nearest neighbor)를 결합시켜서 생긴 것이 아니라 filter window라는 개념에 의해서 확대 및 축소과정에서 생기게 된 경우이며, 배율에 따라서는 <그림 13>과 같은 2 점을 이용해서 보간하는 경우도 생기고, NN과 같이 1점을 이용해서 보간하는 경우도 생기게 된다.

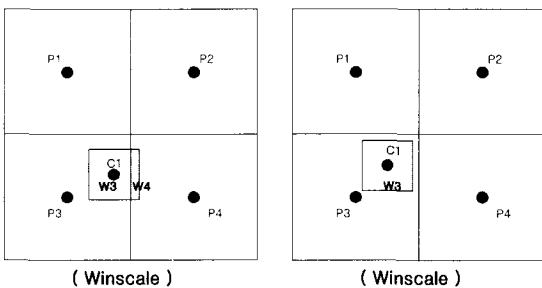


그림 13. bilinear와의 차이
Fig. 13. algorithmic difference between bilinear & winscale.

축소의 경우에도 <그림 14>와 같은 경우가 생기게 되므로 점유비가 bilinear와 다르게 되어서 보간화소의

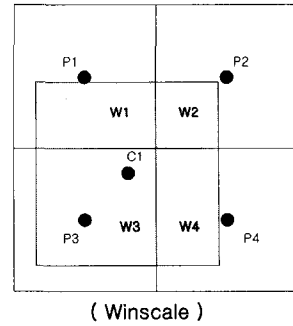


그림 14. bilinear와의 차이 (축소)
Fig. 14. algorithmic difference between bilinear & winscale (image reduction).

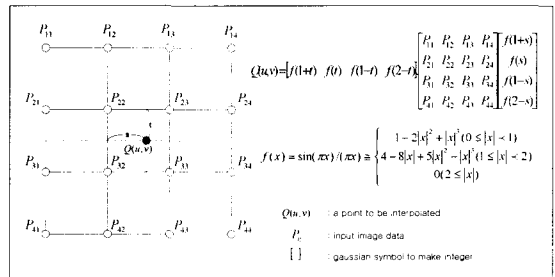


그림 15. bicubic
Fig. 15. bicubic algorithm.

값이 달라지게 된다.

위의 알고리즘 중 가장 복잡한 알고리즘으로 sync 함수를 근사화해서 출력 영상의 화소 값을 구하는데 이용한다. 기본적으로 다른 알고리즘들이 출력 영상의 화소 주변의 4 pixel 을 이용하는 것과는 달리 bicubic 알고리즘은 주위의 16 화소 값을 이용한다. 계산량이 많은 대신 좀더 높은 화질의 영상을 얻을 수 있는 장점을 가지지만 위의 수식에서도 보이듯이 H/W 구현 시 H/W cost가 매우 크다.

앞에서 설명한 비교 알고리즘들과 winscale 알고리즘으로 앞에서 설명한 테스트 방법을 이용해서 얻은 XGA 의 영상에 대한 RGB 채널에 대한 PSNR 값은 <표 2>와 같다.

비교 알고리즘에도 winscale H/W 와 동일한 precision limit 을 주었다. Precision limit 을 주지 않으면 올바른 비교가 되지 않기 때문이다. 그리고 정량적인 비교를 위해서 PSNR 값을 사용하기는 했지만, 통상적으로 512x512 이하 정도의 작은 영상이 아닌 위에서 사용된 큰 영상(1024x768 이나 1280x960)의 경우에는 PSNR값이 정량적인 비교에 결정적인 의미는

지니지 못한다. 그러나 <표 2>에서 보이듯이 다른 알고리즘에 비해 PSNR 향상을 확인할 수 있었다.

표 2. 영상의 PSNR
Table 2. PSNR of Image (RGB channel).

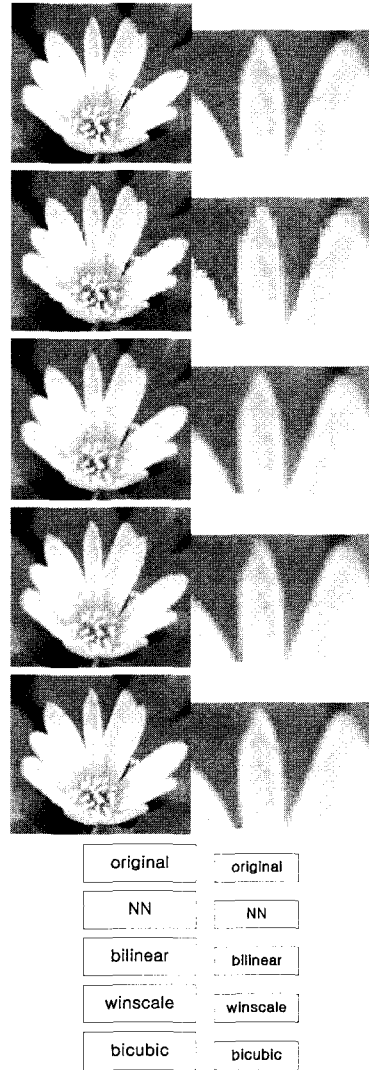
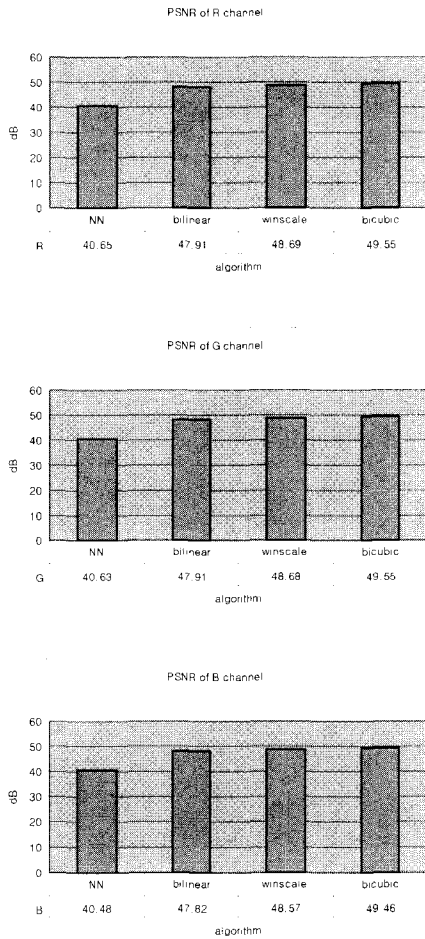


그림 16. 자연 test image
Fig. 16. test image of nature.

그럼 각각의 알고리즘을 이용해서 확대를 수행한 영상의 일부를 확대해서 보도록 하자. 자연영상과 텍스트(arial font)가 나타난 부분을 1024×768의 영상에서 캡처했다. 정확한 PSNR비교를 위해서는 스케일전의 영상과 스케일 후의 영상 모두를 원 영상(original image)을 가지고 있어야 한다. 그런데 각각의 영상에 대한 원 영상을 만들어 내기가 어려워서 원 영상을 축소하고 이를 다시 원 영상과 같은 크기로 확대를 수행한 후 원 영상(original image)과의 비교를 수행했다.

ABCDEFGHIJKLMNOPQRSTUVWXYZ
 ABCDEFGHIJKLMNOPQRSTUVWXYZ
 ABCDEFGHIJKLMNOPQRSTUVWXYZ
 ABCDEFGHIJKLMNOPQRSTUVWXYZ
 ABCDEFGHIJKLMNOPQRSTUVWXYZ



```

ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ

```

A B C

```

ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ

```

A B C

```

ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ

```

A B C

```

ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ

```

A B C

- original
- original
- NN
- NN
- bilinear
- bilinear
- winscale
- winscale
- bicubic
- bicubic

그림 17. text test image
Fig. 17. test image of text.

IV. 결 론

본 논문에서는 winscale 알고리즘을 제안했고, winscale 알고리즘을 이용한 영상 스케일러를 verilog XL 을 이용해서 구현하였다. 설계한 영상 스케일러는 크게 winfilter, prescaler, parameter register, sram interface로 이루어져 있다. gate 수 29000 의 VLSI로 구현되어 칩 테스트 중이며 구현을 위해서 삼성의 0.5um KG80 SOG 라이브러리가 사용되었다. 테스트 칩의 동작 주파수는 25MHz 이며 target LCD-panel 해상도인 640×480에 winscale 알고리즘을 이용해서 영상 스케일링을 해서 디스플레이하게 된다. H/W cost 가 비슷한 기존의 bilinear 알고리즘 보다 높은 화질을 보인다. 그리고 vertical interpolation에 사용되어야 하는 라인 버퍼가 2 줄로 다른 FIR 필터가 들어가는 영상 스케일러보다 라인버퍼의 크기가 compact 한 장점을 지닌다. 향후 진행되어야 할 연구 과제는 winscale 알고리즘의 화소 모델이 사각형인데 실제 human vision에서의 화소 모델은 구형이므로 이러한 human vision에 맞게 화소 모델과 필터 모델을 변형시키는 일이다. 이러한 과정을 통해 좀더 높은 화질의 영상을 얻을 수 있을 것으로 기대된다.

참 고 문 헌

[1] A. K. Jain, Fundamentals of Digital Image Processing, Prentice-Hall, 1989.

- [2] J. S. Lim, Two-dimensional Signal and Image Processing, Prentice-Hall, 1990.
- [3] DAN E.DUDGEON, Multidimensional Digital Signal Processing, Prentice-Hall, 1984.
- [4] A.M.Darwish, M.S.Bedair, and S.I.Shaheen, "Adaptive resampling algorithm for image zooming," IEE Proc.-Vis. Image Signal Process, Vol. 144, No. 4, August 1997.
- [5] Joshua Larson-Mogal, and Silicon Grapics, Inc, "Flat Panel Display Interface Technologies", Silicon Graphics, 1998.
- [6] Passport, "0.6-Micron, 5-Volt, Datapath Library", November 1994.
- [7] Lirida Alves de Barros Naviner, "VLSI ARCHITECTURES FOR READ-TIME MOVING IMAGE RESAMPLING", IEEE, 1996.
- [8] Michael Unser, "Enlargement or Reduction of Digital Images with Minimum Loss of Information", IEEE Trans on Image Processing Vol 4, No 3, March 1995.
- [9] 신정호, 우현배, 강상규, 백준기, "디지털 캠코더용 영상확대 시스템의 설계 및 구현", 제4회 IDEC MPW 발표회 논문집, pp. 170~175
- [10] Byung Ki Park, Chul Ho Choi, Jin Sung Park, "A Universal Controller for a-Ai TFT LCD of SXGA Class", ASIA DISPLAY 98.
- [11] "FPD 상에서 다중 신호원을 디스플레이하기 위한 델타형 보간 알고리즘", 한국 멀티미디어학회 논문지 제2권 1 호 논문번호 990201-

저 자 소 개



成始門(正會員)

1999년 한국과학기술원 전기 및 전자공학과 학사. 2001년 한국과학기술원 전기 및 전자공학과 석사. 2001년~현재 (주)팩&큐리텔 연구원. <주관심분야 : 3D 그래픽스, 이미지 스케일러, 디지털 회로 설계>



金春昊(正會員)

1996년 경북 대학교 전자 공학과 학사. 1999년 한국과학기술원 전기 및 전자공학과 석사. 2000년~현재 한국과학기술원 전기 및 전자공학과 박사. <주관심분야 : 디지털 회로 설계, 이미지프로세싱, 컴퓨터그래픽스>



李珍彦(正會員)

한양 대학교 학사. 한국과학기술원 전기 및 전자공학과 석사. 2000년 한국과학기술원 전기 및 전자공학과 박사. 현재 (주)삼성전자 연구원. <주관심분야 : 3D Computer Graphics>



金利燮(正會員)

1982년 서울대학교 전자공학과 학사. 1986년 Stanford University 전자공학과 석사. 1990년 Stanford University 전자공학과 박사. 1990~1993년 Toshiba Corporation 연구원. 1993년~현재 한국과학기술원 전자전산학과 전기및전자전공 정교수. <주관심분야 : 멀티미디어 VLSI 설계, 저전력 설계, EML 설계, 3D Graphics>