

가중 평균 필터를 이용한 Multimedia TV의  
화질 개선에 관한 연구  
Research for Picture Quality Improvement of Multimedia TV Using  
Weighted Average Filter

강 경 용 (전자과)  
Kyung-Yong Kang (Dept. of Electronics)

KeyWords: Interlaced(비월주사), Non-Interlaced(순차주사), Flickering(깜박거림), PSNR, Readability(가독성), 화소(Pixel), Weighted Average Filter(가중 평균 필터)

ABSTRACT: This paper is about the picture quality improvement of multimedia TV using weighted average filter method. It is necessary for multimedia TV to convert interlaced mode to non-interlaced mode, but the basic change method without filtering decreases recognition of character region in image information. It applied the method of weighted average filtering to improve readability and verified the result through software simulation. The result of this research can be used for the improvement of picture quality of multimedia TV in terms of economic solution.

## 1. 서 론

최근에 멀티미디어(multimedia)는 대부분의 전자 제품에 적용되는 필수적인 기능이 되어가고 있다. 이러한 추세에 따라 컴퓨터, 오디오, TV등 별개의 미디어를 다루던 기기 들의 경계가 점차 모호해지고 있으며 각 제품간의 기능의 융합은 점차 빨라지고 있다. TV와 PC 역시 서로 다른 영역 이었으나 최근 융합된 기능의 제품들이 등장하고 있으며 PC에 TV 수신 기능 및 멀티미디어 기능이 추가된 멀티미디어 PC와 TV에 PC 기능이 추가된 멀티미디어 TV의 두 가지 종류로 나눌 수 있다.

비디오 데이터(video data)의 처리라는 관점에서 TV를 위주로 통합하는 경우 PC의 VGA card의 신호를 TV 화면에 표시하려면 순차 주사 방식에서 비월 주사 방식으로 변환해 주는 것이 필요하다.<sup>(1)</sup> 본 논문은 PC에 사용하는 VGA card의 순차 주사 방식 비디오 데이터를 TV에서 사용하는 비월 주사 방식 비디오 데이터로 변환할 때 발생하는 문제점을 개선하려는 것으로 TV의 NTSC/PAL의 비월 주사 방식으로 데이터를 발생하기 위하여 PC내의 VGA 카드 출력인 순차 주사 방식의 데이터를 비월 주사 형식으로 변환할 때 발생하는 화면의 깜박거림(Flickering)과 문자의 경계 영역의 평탄화로 인한 문자의 흐려짐(Smoothing)을 최대한 감소하려는 것이다.

VGA에서 사용하는 525개의 주사선을 TV에서 사용되는 262.5개의 주사선으로 변환하는 과정에서 필터를 사용하지 않는 기본적인 변환 방식을 적용할 때 TV상에 표현되는 문자의 경우에는 문자의 수직 천이 경계(vertical edge transition)영역의 윤곽이 흐려져 작은 글자이거나 굴곡

이 많은 문자인 경우는 문자의 정보성 즉 가독성(readability)이 떨어지게 된다. 문자 정보는 영상 정보와 다르게 정보 전달이 제대로 되지 않는 것은 치명적이기 때문에 VGA 신호를 TV 신호로 변환했을 때 문자의 윤곽이 흐려지는 것을 보완해야 한다. 이에 대한 보완 방법으로는 문자 정보와 그래픽 정보가 혼합되어있는 화면에서 문자의 수직 천이 경계 영역을 분리해야 하고 이들 VGA 신호가 TV 신호로 변환 될 때, 문자의 수직 천이 경계 영역에서 윤곽이 흐려지지 않는 선택적인 필터링 방식을 실행하는 실시간의 동작이 필요하다.

본 논문에서는 문자의 수직 천이 경계를 검출하는 알고리즘을 개발하고 이를 실시간에 영역별로 서로 다른 필터링 방식을 실행 할수 있는 하드웨어를 최종적으로 구현하였으며 문자의 수직 천이 경계 영역의 검출 알고리즘은 히스토그램(histogram)을 이용한 문자 영역에 대한 특성, gradient의 크기를 이용한 천이 검출<sup>(4)(5)</sup> 방법과 지속 길이(run-length)의 제한법 등을 기초로 하였다.

변환에 따른 화질의 열화와 이의 개선을 위하여 문자영역의 수직 천이 경계를 검출하는 알고리즘을 제안하고 소프트웨어적인 방법으로 검증하였으며 이를 기반으로 FPGA 형태의 하드웨어로 구현하였다. 하드웨어로 구현 시 2 라인과 3 라인을 이용하는 경우와 R.G.B의 3가지의 입력과 G 하나의 입력만을 사용하는 경우를 각 각 고려하였고 그 결과를 검출에 필요한 수행시간과 시험화면에 대한 PSNR로 평가하였다.

PC와 TV가 융합되는 추세와 www과 같은 인터넷상의 서비스에 대한 소비자들의 폭발적인 요구를 감안할 때 PC와 TV의 기능이 융합된 멀티미디어 TV가 사용자에게 익숙한 interface를 제공 할 수 있으며 이러한 장점으로 인하여 인터넷과 연관된 새로운 수요가 형성될 것이 기대된다. 이런 경우 본 연구의 결과는 멀티미디어 TV에서 문자가 포함된 화질 개선을 위해 활용 될 수 있을 것이다.

## 2. 순차-비월 주사 변환(noninterlaced-to-interlaced conversion)

순차 주사(noninterlaced) 방식에서 비월 주사(interlaced) 방식으로 변환하는 데에는, 기본 방식(basic noninterlaced-to-interlaced conversion) 필터를 이용하지 않고 단순히 필요한 라인들을 추출하는 방식)과 가중 평균 필터(weighted averaging filter)를 이용하는 방식(noninterlaced-to-interlaced conversion using filtering)의 두 가지 방식이 있다.<sup>(1)</sup>

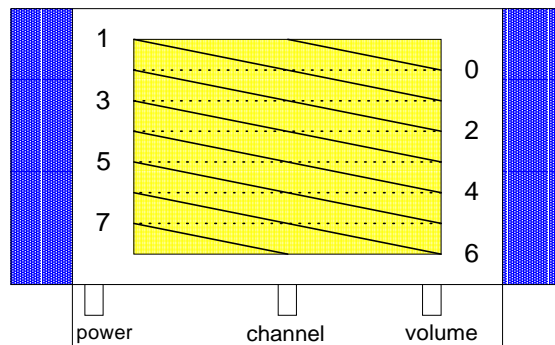
순차 주사 방식의 프레임(frame) 전체에 가중 평균 필터를 이용하는 방식만을 적용하여 변환하는 경우 문자의 수직 천이 경계(vertical edge transition)영역에서 문자의 판독이 어렵도록 흐려지는 문제가 발생하고 프레임 전체에 기본 방식만을 적용하여 변환할 경우는 화면의 깜박거림(flickering)이 심하게 나타난다. 따라서 순차 주사 방식에서 비월 주사 방식으로 변환을 할 때 문자의 경계 영역에 해당하는 부분과 문자의 경계가 아닌 영역을 검출하여 문자의 가독성을 높이도록 각자의 영역에 각기 다른 방식으로 변환할 필요가 있다.

### 2.1 비월 주사 방식

비월 주사 방식은 두 삽입 필드(interleaved field)를 사용하여 하나의 비디오 프레임(video frame)을 완성하는 방식이다.<sup>(1)</sup> 따라서, 한 필드의 라인(line) 수는 한 프레임의 라인 수의 절반이 된다. NTSC 방식의 경우, 프레임 당 525 라인이 있고, 필드 당 262.5 라인이 있다. 비디오 프레임의 홀수 번째 라인들이 한 필드를 구성하고, 짝수 번째 라인들이 다른 한 필드를 구성

하며, 두 필드가 서로 엮혀진다(interlaced). 그림1과 같이 한 필드에 해당하는 홀수의 라인 1,3,5,7..가 화면에 주사되고, 연속적으로 다른 필드의 라인 2,4,8,10...가 주사되어 한 프레임을 완성한다.

비월 주사 방식을 사용하는 이유는 화면의 갱신 속도(Refresh Rate)와 관계가 있다. 초당 30 프레임의 속도로 화면을 갱신하면 다음 프레임이 나오기 전에 이전 프레임의 잔상이 사라져 시청자가 깜박거림을 느끼게 된다. 그러나, 한 프레임을 절반의 정보를 담은 두 필드로 나누어, 초당 60 필드의 속도로 필드 단위로 화면을 갱신하면, 시청자는 연속적이고 부드러운 화면으로 느끼게 된다.



Interlace: field 1 (lines 1,3,5,7) is scanned, then field 2 (lines 0, 2, 4, 6).

그림 1. 비월 주사 방식

Fig. 1 Interlaced Mode

## 2.2 순차-비월 주사 변환

PC는 순차 주사 방식의 디스플레이(display)를 사용하므로 이로부터 비월 주사 방식의 NTSC 비디오 신호를 얻기 위해서는, PC의 VGA 비디오 데이터를 일차적으로 비월 주사 방식으로 변환하는 것이 필요하다.<sup>[1]</sup>

순차 주사 방식에서 비월 주사 방식으로 변환하는 방식 중 기본 변환 방식은 그림 2와 같다. 기본 방식은 순차 주사 방식 비디오 프레임의 홀수 번째 라인들만으로 한 필드를 구성하고 짝수 번째 라인들만으로 다른 한 필드를 구성하는 방식이다. 이 방식은 경제적이고 구현이 쉽지만 문제점을 지니고 있는데, 특히 변환하려는 순차 주사 프레임에 굵기가 한 라인이나 두 라인에 해당하는 긴 가로 방향의 선이 있을 경우 문제가 발생한다.<sup>[1]</sup>

순차 주사 프레임에 굵기가 한 라인에 해당하는 긴 가로 방향의 선이 하나 있을 때, 이 선은 변환 후 1초에 30번씩 디스플레이 되므로 (NTSC의 frame refresh rate가 30Hz, field refresh rate가 60Hz인데, 이 선은 그림 2의 #1 필드와 #2 필드 중 한 필드에서만 나타나게 된다.) 깜빡거리게 된다. 그리고, 순차 주사 프레임에 굵기가 두 라인에 해당하는 가로 방향의 긴 선이 있을 경우, 이 선을 구성하는 두 라인은 변환 후에 서로 다른 필드에서 교대로 디스플레이

되므로 위아래로 진동하게 된다. 깜박거림 효과는 물체(object)의 상하 경계에서도 나타날 수 있어서, 순차 주사 프레임 전체를 기본 변환 방식만으로 변환할 경우 화면 전체에서 심한 깜박거림을 느끼게 되는 반면에 수직 천이 경계의 평탄화(smoothing out) 현상이 없으므로, 문자가 선명하게 나타난다. 기본적인 변환 방식은 문자의 수직 천이 경계가 선명하게 나타나는 반면에, 깜박거림이 심하다는 단점을 가지고 있다.

순차 주사 방식에서 비월 주사 방식으로 변환하는 방식 중 가중 평균 필터를 이용하는 방식은 그림3과 같으며 세 주사 라인의 순차 주사 데이터를 이용하여 한 주사 라인의 비월 주사 데이터를 만들어 낸다. 세 주사 라인의 순차 주사 데이터를 가중 평균 필터링 하여 한 주사 라인의 비월 주사 데이터를 만들어 내므로, 순차 주사 프레임에서의 급격한 수직 경계 천이가 몇 라인의 비월 주사 라인에 걸쳐 평탄화(smoothing out)된다.<sup>[1]</sup> 순차 주사 프레임에서의 한 라인 굵기의 긴 수평 방향 선의 경우, 가중 평균 필터를 이용하여 변환하면 몇 라인의 비월 주사 라인에 걸쳐 평탄화되기 때문에 변환 후 깜박거림이 나타나지 않는다.

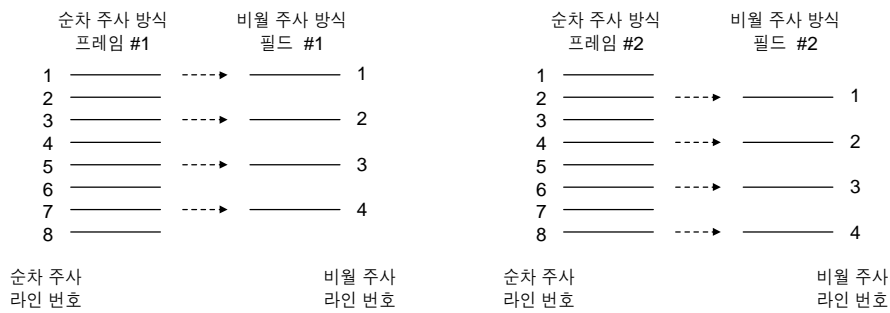


그림 2. 기본적인 변환 방식

Fig. 2 Basic Conversion Method

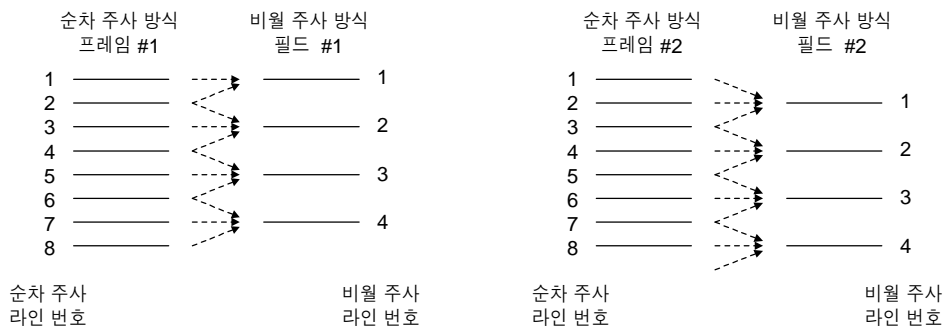


그림 3. 가중 평균 필터를 이용하는 방식

Fig. 3 Weighted Average Filter Method

필터를 이용하는 방식은 깜박거림이 나타나지 않는다는 장점이 있는 반면에, 문자의 수직 천이 경계가 상하로 퍼져 문자를 알아보기 어렵게 된다는 단점이 있다. 가중 평균 필터를 이용하는 방식은 각 비디오 성분(여기서는 R, G, B)에 대한 비디오 데이터를 변환에 쓰일 때까지 저장할 라인 저장 장치(line store)와 가중 평균 연산을 행할 디지털 필터(digital filter)를 필요로 한다. 그림 4는 가중 평균 필터를 이용하여 변환하는 예를 G성분에 대하여 나타내고 있다.

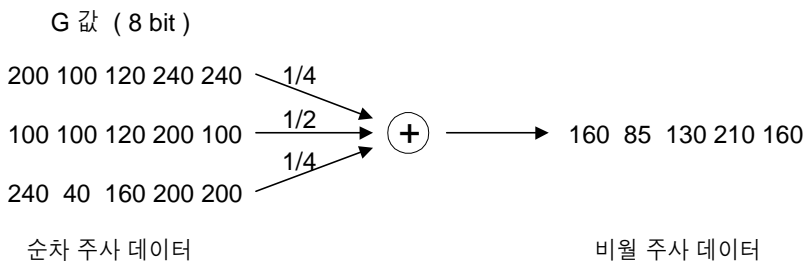


그림 4. 1:2:1 가중 평균 필터를 이용한 변환의 예

Fig. 4 The Example of Conversion using Weighted Average Filter

### 2.3 순차-비월 주사 변환시의 문제점

앞의 두 절에서 기술한 바와 같이, 순차 주사 방식에서 비월 주사 방식의 변환은 기본 방식과 가중 평균 필터를 이용하는 방식의 두 가지 방식이 있다. 순차 주사 방식의 비디오 프레임 전체에 가중 평균 필터를 이용하는 방식만을 적용하여 변환을 할 경우 문자의 수직 천이 경계가 문자의 판독이 어렵도록 흐려지는 문제가 발생하고 프레임 전체에 기본 방식만을 적용하여 변환하는 경우 화면의 깜박거림이 심하게 나타난다. 따라서 순차 주사 방식에서 비월 주사 방식으로 변환하는 문자의 경계인 영역과 문자의 경계가 아닌 영역을 서로 다른 변환 방식으로 변환해 줄 필요가 있다.

CH7001 chip은 VGA to NTSC/PAL encoder로서 순차-비월 주사 변환을 행하는 기능 블록(function block)을 포함하고 있다. 주사 방식 변환을 행할 때 사용하는 가중 평균 필터를 4가지 모드를 갖추고 있으며 네 가지 가중 평균 필터는 외부에서 선택이 가능하다. CH7001의 입력은 VGA card의 RGB output이고, 출력은 NTSC/PAL S-Video output이다.<sup>[6]</sup> 이와 같은 기능을 제공해 주는 CH7001을 이용한 하드웨어로 순차-비월 주사 방식 변환시의 문제점을 개선할 수 있다.

CH7001은, MS0, MS1 pin을 통하여, 주사 방식 변환시의 가중 평균 필터 방식을 외부에서 선택하는 것이 가능하며,<sup>[6]</sup> 4 가지 필터 방식은 표 1과 같다.

표 1에서, 0:1:0 가중 평균 필터 방식은 기본적인 변환 방식과 결과적으로 같은 방식이며 따

라서, 0:1:0 가중 평균 필터 방식은 다른 3 가지의 가중 평균 필터 방식에 비해 현저하게 깜박 거림이 심한 반면, 물체의 수직 천이 경계는 단연 선명하다. 나머지 세 가지 필터를 비교하자면, 1:2:1 가중 평균 필터 방식이 가장 깜박거림이 없으면서 물체의 수직 천이 경계가 가장 흐릿하고 할 수 있으나, 세 가지 방식 사이에는 큰 차이는 없다.

CH7001 chip을 이용하여 순차 주사 프레임을 비월 주사 필드로 변환할 때, 순차 주사 프레임 전체에 0:1:0 가중 평균 필터 방식만을 적용하면 수직 천이 경계 부분이 선명해져 깜박거림이 심하게 되고, 프레임 전체에 다른 가중 평균 필터 방식만을 적용하면 화면 전체가 깜박거림은 없지만 수직 천이 경계가 흐릿하게 퍼져 문자를 알아보기 어렵게 된다.

CH7001의 R, G, B input pin으로 들어가는 VGA card의 아날로그(analog) R, G, B 신호를 CH7001 외부의 A/D 변환기(converter)에도 입력시켜 A/D 변환을 하고 이렇게 얻은 R, G, B 값은 순차-비월 주사 변환되기 위해 CH7001에 들어간 화소의 R, G, B 값이다.

표 1. CH7001의 4가지 필터링 방법

Table1. Four weighted average filter method of CH7001

Filter Mode	flickering	수직 천이 경계의 선명도
0:1:0 averaging	아주 심함	아주 선명
1:3:1 averaging	조금 있음	더 선명
1:2:1 averaging	거의 없음	아주 흐림
1:1:0 averaging	조금 있음	흐림

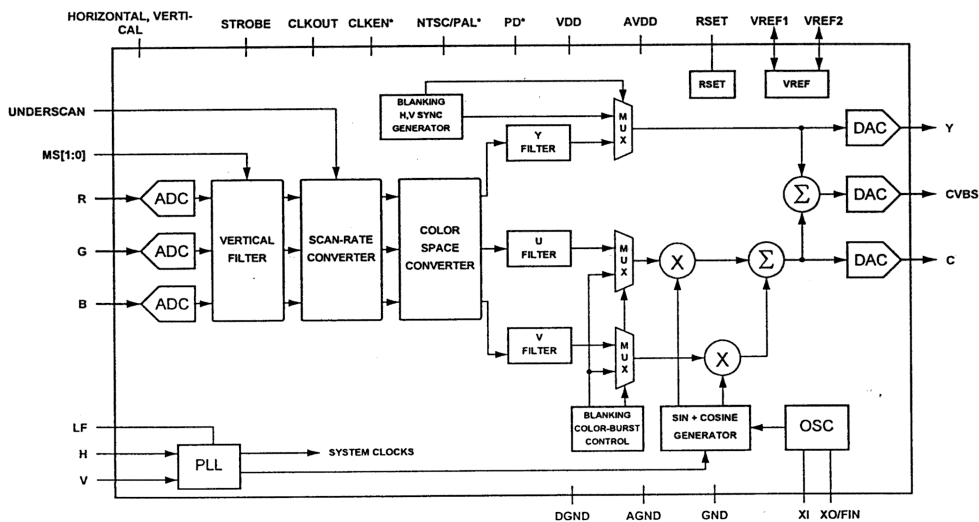


그림 5 CH7001의 기능 블록 다이어그램

Fig. 5 Functional block diagram of CH7001

이 값을 이용하여 해당하는 화소가 문자의 수직 천이 경계 영역인지 판명하고 그에 따라 chip 을 제어하여 0:1:0 방식의 필터링으로 순차-비월 주사 변환을 하여 문자의 수직 천이 경계가 선 명하게 보일 수 있도록 한다. 화소가 문자의 수직 천이 경계가 아닌 것으로 판명되면 MS0, 1 pin을 제어해 감박거림이 적은 방식으로 순차-비월 주사 변환되도록 한다.

이를 실제 하드웨어로 구현할 때는 G 신호만을 A/D 변환해서 화소의 G 값을 이용하여 VGA card에서 출력되는 화소가 수직 천이 경계인지 판명하고 하드웨어에서의 실제 수직 천이 경계 검출은 이전 라인에 해당하는 화소들의 G 값들을 한 개의 line memory에 저장해 놓고, 현재 VGA card에서 출력되는 화소의 G 값을 이와 비교하여 이루어지며 제어부는 FPGA의 일 종인 Altera EPLD(EPM7128)로 구현하였다.

그림 6은 본 절에서 기술한 하드웨어의 블록 다이어그램(block diagram)을 나타내고 있다.

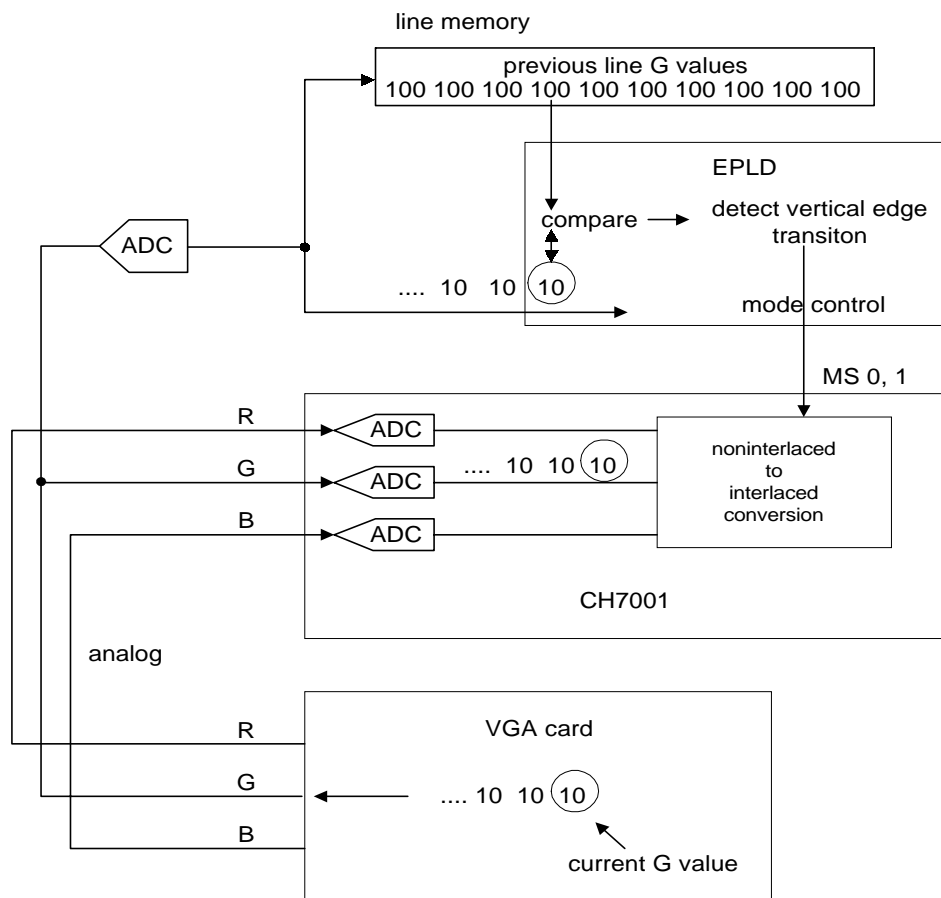


그림 6. 변환 제어 하드웨어의 블록 다이어그램  
Fig. 6 Block Diagram of Conversion Control Hardware

### 3. 영상의 라인을 이용한 수직 천이 경계 검출 알고리즘

영상의 이전 라인의 화소의 값과 현재의 화소의 값을 이용하여 수직 천이 경계를 검출하는 알

고리즘을 제안하며 그림 7에서 pixel(x, y)는 좌표 (x, y)에 해당하는 화소를 나타내고, pixel(x, y-1)은 pixel(x, y)와 x축 상의 위치가 같은 그 이전 라인의 화소를 나타내며 수직 천이 경계를 검출하기 위하여 아래와 같이 히스토그램과 gradient의 특성을 이용하였다.

### 3.1 히스토그램을 이용한 영역의 특성 파악

영상의 한 라인에 대하여 히스토그램을 작성하고, 작성된 히스토그램을 분석하여, 라인의 특성을 파악한다.<sup>[2][3]</sup> 이 라인의 특성이 다음 라인에서도 지속된다는 가정하에 새로운 라인에서의 수직 천이 경계 검출에 이전 라인에서 얻어진 히스토그램을 이용하도록 한다.

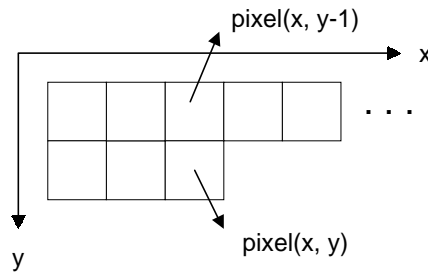


그림 7. 알고리즘 기술을 위한 notation

Fig. 7 Notation of Algorithm Description

영상의 한 라인에 대한 각 화소에 gray level에 대한 히스토그램은 하드웨어의 구현를 고려하여 메모리 요구량을 줄이도록 해야한다. 한 라인의 각 화소에 대하여 histogram array의 frequency field를 계속 갱신하다가 그 라인이 끝나면 히스토그램을 분석하여 그 라인의 영역의 형태와 그 라인에서의 문자의 gray level을 알아낸다. 한 라인의 특성이 그 다음 라인까지 지속되는 것으로 가정하여 히스토그램을 분석하여 수직 천이 경계 검출에 이용한다.

### 3.2 Gradient를 이용한 수직 천이 경계의 검출

영상을 나타내는 함수  $f(x, y)$ 에 대하여,  $f(x, y)$ 의 gradient는

$$\nabla f(x, y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (1)$$

로 정의된다. 경계 검출에 중요한 양은 gradient의 크기로,

$$|\nabla f(x, y)| = [G_x^2 + G_y^2]^{\frac{1}{2}} \quad (2)$$

과 같이 표현되며,  $f(x, y)$ 의  $\nabla f(x, y)$  방향으로의 단위 길이당 최대 증가율을 나타낸다.<sup>[4][5]</sup>

$|\nabla f(x, y)|$ 는  $|G_x| + |G_y|$ 로 근사화될 수 있는데, 수직 천이 경계만을 검출하는 경우,  $|G_y|$ 만을 사용하여  $|G_y|$ 가 일정 threshold이상일 때 (x, y)지점을 수직 천이 경계로 검



출할 수 있다.

$$|G_y| \approx \left| \frac{f(x, y+\Delta y) - f(x, y)}{\Delta y} \right| \quad (3)$$

이므로, 디지털 영상에서는

$$|G_y| \approx |G_y(x, y)| = |g(\text{pixel}(x, y)) - g(\text{pixel}(x, y-1))| \quad (4)$$

로 근사화될 수 있다. 여기에서  $g(\text{pixel}(x, y))$ 는  $\text{pixel}(x, y)$ 의 gray level을 나타내고  $|G_y(x, y)|$ 가 일정 threshold 이상일 경우  $\text{pixel}(x, y)$ 를 수직 천이 경계로 검출할 수 있다. 실험에 사용하는 영상의 문자의 지속 길이(run-length)는 평균적으로 10 화소 안팎이며, 10 화소가 넘는 큰 문자의 경우 10 화소를 넘어서는 부분부터는 0:1:0 가중 평균 필터링을 적용하지 않아도 문자를 충분히 식별할 수 있다. 연속되는 긴 라인의 경우는 문자와 구별하여 10 화소를 넘어서는 부분부터 1:2:1 방식으로 필터링 하여 화면 전체의 깜박거림(Flickering)이 감소하도록 한다.

### 3.4 통합 알고리즘

앞 절에서 언급한 히스토그램과 gradient의 특성을 함께 사용하는 통합 알고리즘의 대략적인 내용은  $\text{pixel}(x, y-1)$ 이 속해 있는 라인의 히스토그램으로부터 얻은 case flag 값, character 값을 바탕으로 하여,  $\text{pixel}(x, y)$ 의 시점에서 수직 천이 경계를 검출하는 것이다.

이전 라인의 히스토그램을 분석하여 얻은 character 값의  $\pm 25$  범위 내에 실제 문자의 gray level이 존재할 것이라는 것과, 경계에서는 gradient의 크기가 일정 threshold 이상이라는 것을 이용한 것이며 이전 라인에서 다수를 차지하는 문자의 gray level과는 다른 gray level의 문자가 있을 경우, 그 문자의 수직 천이 경계는 검출이 안되기 때문에, 그러한 문자의 수직 천이 경계를 검출하기 위한 것이다.

VGA 순차 주사 비디오 프레임의 2 라인을 이용하여 수직 천이 경계를 검출하는 알고리즘을, 적용한 경계 검출 하드웨어는 그림 8과 같다.

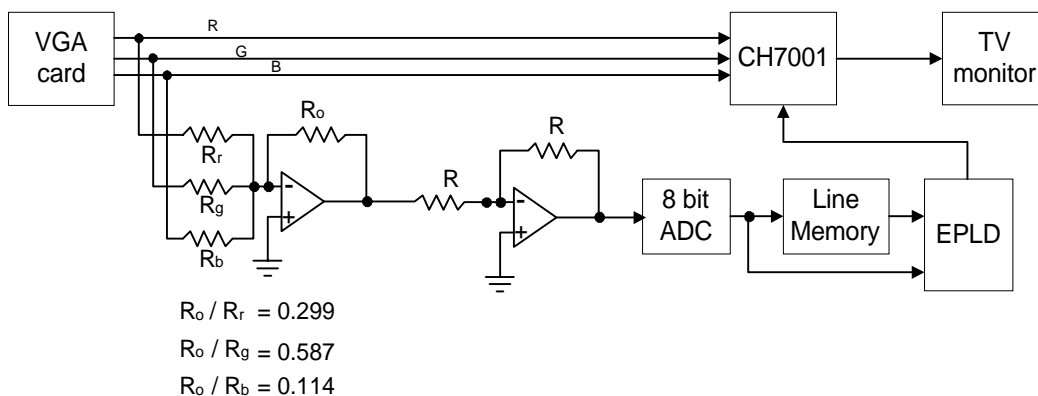


그림 8. 2 라인을 이용한 경계 검출 하드웨어 구조

Fig. 8 Hardware Architecture of Edge Detection using 2 lines

#### 4. 소프트웨어를 이용한 알고리즘 검증

3장에서 제안한 2 라인을 이용한 수직 천이 경계 검출 알고리즘을 검증하기 위한 소프트웨어 검증(software simulation)은 실험 대상으로 사용할 영상을 선정하고 실험 대상으로 선정된 영상에, 제안된 수직 천이 경계 알고리즘을 적용하여 경계를 검출한다.

실험 영상의 영역별로 PC의 병렬 포트(parallel port)를 통해 제어하고 문자의 수직 천이 경계 영역에는 0:1:0 필터링 방식이 적용되도록 하고, 그 이외의 영역에는 흐릿한 필터링 방식이 적용되도록 하여, TV 화면에서 전체적으로 깜박거림(Flickering)이 줄고 문자는 명확히 보이도록 하고 이를 모니터를 통하여 확인한다.

실험에는 C 언어를 사용하여 VGA card의 register를 직접 제어하거나 VGA ROM BIOS를 이용하는 방식으로 작성하였다.<sup>(7)(8)</sup> CH7001 chip을 제어하는 부분은 빠른 속도가 요구되어 BORLAND C++에서 제공하는 inline assembly<sup>(9)</sup>를 이용하였다.

##### 4.1 실험 영상

실험 대상 영상으로는 다음의 4가지 영상을 사용하였다.

Window-95의 file 탐색 화면, DOS prompt상에서의 color 화면, DOS prompt상에서의 흑백 화면, 긴 라인과 'ㄹ'자 및 여러 가지 문자들을 섞어놓은 화면이며 그림 10에서 긴 라인은 0:1:0 변환 방식에서 깜박거림이 두드러지게 나타나는 패턴이고, 'ㄹ'자는 흐릿한 방식의 변환을 하면 수직 천이 경계가 평탄화(smoothed out)되어서 인식할 수 없는 대표적인 패턴이다. 따라서, 실험 영상 4를 긴 라인 영역은 경계가 평탄화되는 방식으로, 'ㄹ'자 및 기타 문자 영역은 0:1:0 방식으로 변환하면, 깜박거림이 적으면서 문자는 선명한 화면을 얻을 수 있다.

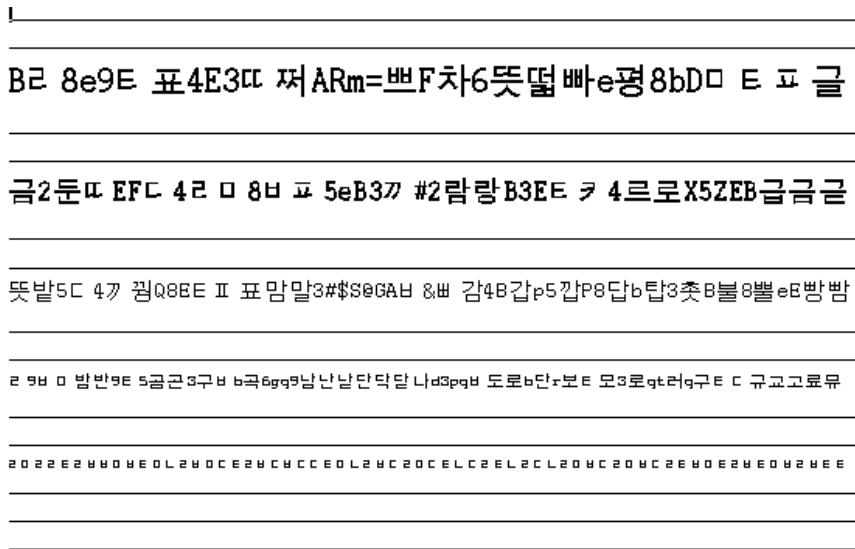


그림 10. 긴 라인과 및 여러 가지 문자를 섞은 영상

Fig. 10 Long line and Image of Mixed with Different Characters

실험 대상 영상은 모두 PCX file의 형태이다. 실험 대상 PCX file을 각 화소에 대한 gray level을 나타내는 file로 변환하고, 3장에서 제안된 알고리즘을 적용하여 수직 천이 경계를 검출한다.

#### 4.2 수직 천이 경계 검출 알고리즘의 적용

하드웨어 구현 시의 제한 조건(하나의 line memory 사용)을 그대로 적용하여 이전 라인 하나의 gray level 값들을 저장할 수 있는 array를 설정하고 그 것과 현재 화소의 gray level 값만을 이용하여 수직 천이 경계를 검출하였다. gray level을 구할 때, R, G, B 값으로부터 gray scale 변환하여 얻은 것과 G 값을 그대로 대치한 것의 두 가지를 사용하여 실험을 하였다. 수직 천이 경계로 검출된 화소에 대하여서는 (x, y) 좌표를 file에 저장하였고, 시각적인 확인을 위해서는 경계로 검출된 화소에 일정한 색깔을 출력하는 방법과 VGA card의 DAC color register를 제어하여 경계에 해당하는 화소가 blinking되게 하는 두 가지의 방법을 사용하였다.<sup>[7]</sup> 경계로 검출된 화소에 일정한 색깔을 출력한 경우, Paint Brush program의 확대 기능을 이용하여 화소 단위로 경계 검출 여부를 확인할 수 있다.

Table 2. Vertical Transition edge detection  
수직 천이 경계 검출률 [%]

	file manager	DOS color	DOS B/W	평균
G	100	98.30	100	99.43
R, G, B	100	99.94	100	99.98

표 2는 실험 대상 화면에 대하여, R, G, B 값을 모두 사용한 경우와 G 값만을 사용한 경우의 문자의 수직 천이 경계 검출 정도를 비교한 것이다. G 값만을 사용하여 경계를 검출하는 경우, G 값이 작은 비율을 차지하거나 0인 색깔의 문자가 포함되어 있는 DOS prompt color 화면은, file manager 화면이나 DOS prompt B/W 화면에 비하여 검출률이 떨어지는 것을 볼 수 있다. G 값이 작은 비율을 차지하는 문자의 비율이 증가하면, 이에 반비례하여 검출률은 더 감소할 것이다.

#### 5. 영상의 3 라인을 이용한 수직 천이 경계 검출 알고리즘

3장에서 이전의 한 라인과 현재의 한 라인, 즉 2 라인을 이용하여 문자의 수직 천이 경계를 검출하는 알고리즘을 제시하였다. 본 장에서는, 한 라인을 더 이용하여, 이전의 두 라인과 현재의 한 라인으로 문자의 수직 천이 경계를 검출하는 알고리즘을 제시하고자 한다. 하드웨어의 구현

측면에서 본다면, 2 라인을 이용하는 알고리즘은 하나의 line memory를 사용하여 구현이 가능하고, 3 라인을 이용하는 알고리즘은 2개의 line memory를 사용하여 구현이 가능하다.

3 라인을 이용하여 수직 천이 경계를 검출하는 알고리즘을 하드웨어로 구현을 할 때, pixel(x, y)는 시간상 현재의 화소에 해당한다. pixel(x, y)의 gray level 값은 현재 화소의 R, G, B 값을 각각 A/D 변환한 후, 이를 gray scale 변환하여 얻고, pixel(x, y-1)과 pixel(x, y-2)의 gray level 값은 각각 line memory 1과 line memory 2에 저장되어 있는 것을 읽어 들인다. 3 라인을 이용하는 알고리즘은, 시간상 현재의 화소가 pixel(x, y)인데 반하여, 시간상 과거의 화소인 pixel(x, y-1)에 대하여 경계 여부를 판단하도록 되어있다. 따라서, pixel(x, y-1)이 CH7001 chip내에서 실제로 가중 평균 필터링이 되는 시간에 MS0, MS1 pin에 필터 방식 제어 신호를 보내기 위해서는, 그림 11과 같이, CH7001 chip으로 들어가는 R, G, B 신호를 지연시킬 필요가 있다.

2 라인을 이용한 경계 검출 알고리즘과 3 라인을 이용한 경계 검출 알고리즘을 실험 대상 영상에 적용하여 R, G, B 값을 모두 사용하여 경계를 검출한 결과를 2 라인을 이용한 알고리즘과 3 라인을 이용한 알고리즘의 경계 검출 성능에는 거의 차이가 없는 것을 확인할 수 있다.

경계 검출 정도에는 차이가 없으나, 3 라인을 이용한 알고리즘은 2 라인을 이용한 알고리즘에 비하여 간단하기 때문에, 검출에 걸리는 시간이 더 짧다는 장점이 있다. 그림 11은, 2 라인을 이용한 경계 검출 알고리즘과 3 라인을 이용한 경계 검출 알고리즘을 실행하는 데 소요되는 시간을 비교하여 보여주고 있다.

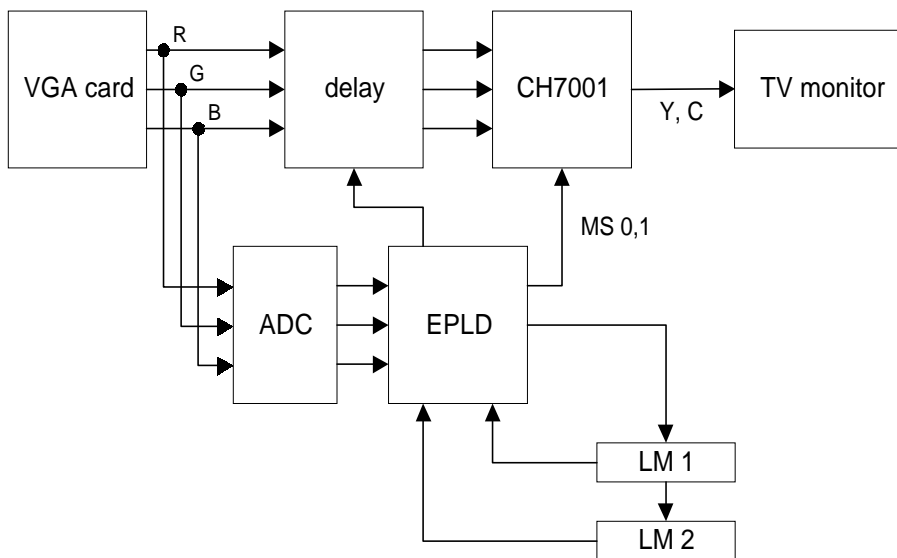
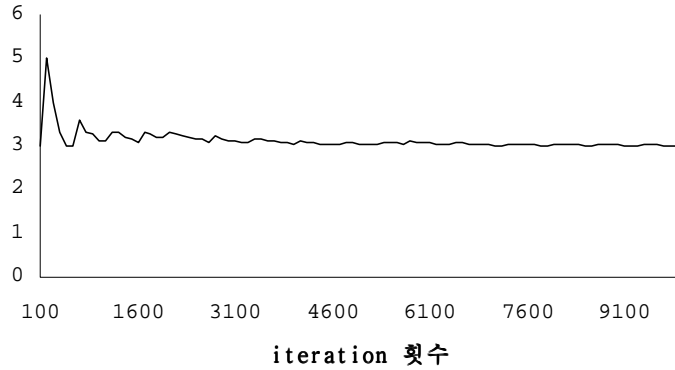


그림 11 영상의 2 라인을 이용한 알고리즘  
Fig. 11 2 lines Algorithm



(2 라인 알고리즘의 실행 시간) / (3 라인 알고리즘의 실행 시간)

그림 12 2 라인 알고리즘과 3 라인 알고리즘의 실행 시간 비교

Fig.12 Comparision of Execution Time 2 Lines And 3 Lines

6. 경계 검출을 위해 R, G, B 값을 모두 사용하는 경우와 G 값만을 사용하는 경우의 비교

본 장에서는, 문자의 수직 천이 경계를 검출할 때, G 값만을 사용하여 경계를 검출하는 경우와 R, G, B 값을 모두 사용하여 경계를 검출하는 경우를 비교하고자 한다. 그림 13은 G 값만을 사용하여 경계를 검출하는 하드웨어 구조를 나타내고 있고, 그림 14는 R, G, B 값을 모두 사용하여 경계를 검출하는 하드웨어 구조를 나타내고 있다.

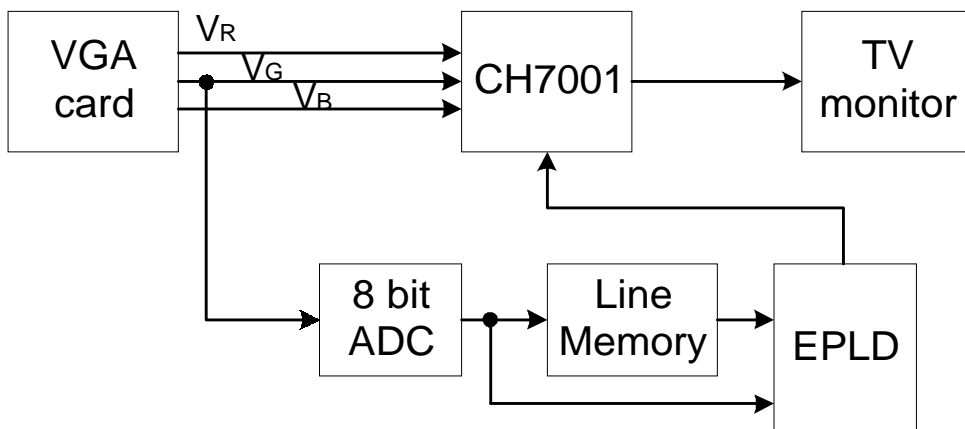


그림 13. G 값만을 사용하는 하드웨어 구조

Fig. 13 Hardware Architecture using only G Value

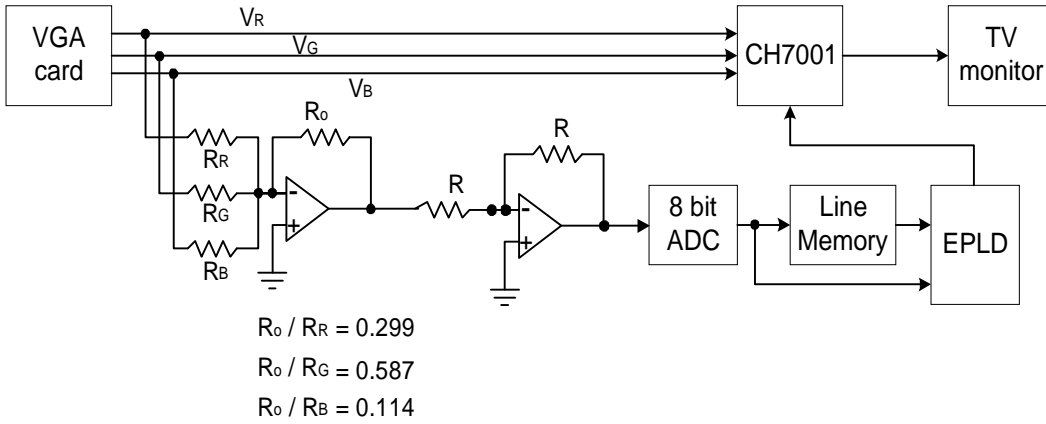


그림 14 R, G, B 값을 모두 사용하는 하드웨어 구조

Fig. 13 Hardware Architecture using R, G, B Values

그림 13, 14에서 VGA card의 출력 R, G, B 값을  $V_R$ ,  $V_G$ ,  $V_B$ 로 표현한 것은 그 값들이 아날로그 전압 값들임을 표시하기 위해서이다. 즉,

$$R = \text{ADC} [ V_R ], G = \text{ADC} [ V_G ], B = \text{ADC} [ V_B ] \quad (5)$$

의 관계가 성립한다. 여기에서,  $\text{ADC} [ ]$ 는 A/D 변환 operation을 나타내며, A/D 변환 operation의 구체적 내용은 식 6과 같다.

$$\text{ADC} [ V_{in} ] = i, \quad (6)$$

$$i \text{는 } \frac{0.714}{256} i \leq V_{in} < \frac{0.714}{256} (i+1) \text{인 정수,}$$

$$0 \leq i \leq 255, 0 \leq V_{in} < 0.714$$

실험 대상 영상 파일이 각 화소에 대하여 R, G, B 각각 8 bit을 할당하고 있기 때문에, 그림 6-1의 하드웨어 구조에서 8 bit ADC의 출력값은 실험 대상 영상 파일의 각 화소에 대한 G 값과 같다고 할 수 있다. 그림 14에서 두 번째 op amp 출력단의 전압이

$$R_0(V_R/R_R + V_G/R_G + V_B/R_B) = 0.299 V_R + 0.587 V_G + 0.114 V_B \quad (7)$$

가 되므로, 8 bit ADC의 출력값은

$$Y_1 = \text{ADC} [ 0.299 V_R + 0.587 V_G + 0.114 V_B ] \quad (8)$$

가 된다. 한편, software simulation에서 각 화소의 R, G, B 값( 각 8 bit )에 대하여 행하는 gray scale 변환은 아래와 같다.

$$\begin{aligned}
 Y_2 &= \text{ROUND} [ 0.299 R + 0.587 G + 0.114 B ] \quad (9) \\
 &= \text{ROUND} [ 0.299 \text{ADC} [ V_R ] + 0.587 \text{ADC} [ V_G ] + 0.114 \text{ADC} [ V_B ] ],
 \end{aligned}$$

## ROUND [ ] : 반올림 연산

G 값만 사용하는 하드웨어에서의 ADC 출력은 실험 대상 영상 파일의 G 값으로 대체될 수 있고, R, G, B 값을 모두 사용하는 하드웨어의 ADC 출력은 실험 대상 영상 파일의 R, G, B 값을 gray scale 변환한 것으로 대체될 수 있다. G 값만 사용하는 하드웨어와 R, G, B 값을 모두 사용하는 하드웨어의 성능 비교가, 실험 대상 영상의 G 값만 사용하는 software simulation과 R, G, B 값을 gray scale 변환하여 사용하는 software simulation의 성능 비교로 대체될 수 있는 것이다.

### 6.1 PSNR의 측정

PSNR(Peak Signal to Noise Ratio)<sup>(4)(5)</sup>은 일반적으로 영상의 질에 대한 객관적인 평가를 위해 사용되며 그 식은 다음과 같다.

$$\text{PSNR} = 10 \cdot \log_{10} \frac{(255)^2}{\text{MSE}} \quad [\text{dB}] \quad (10)$$

$$\text{MSE} = \frac{1}{NM} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [f(x, y) - \hat{f}(x, y)]^2 \quad (11)$$

위 식에서 MSE는 Mean Square Error를 나타내고,  $f(x, y)$ 는 원래의 영상을,  $\hat{f}(x, y)$ 는 처리된 영상을 나타내며,  $N \times M$ 은 영상의 크기를 나타낸다. 본 실험에서는, 실험 대상 영상 파일의 R, G, B 값을 gray scale 변환한 것을  $f(x, y)$ 로, 실험 대상 영상 파일의 G 값을  $\hat{f}(x, y)$ 로 설정하여 PSNR을 측정하였다. gray scale 변환에 사용한 식은 다음과 같다.

$$Y = \text{ROUND} [ 0.299 R + 0.587 G + 0.114 B ], \quad (12)$$

ROUND [ x ] : x를 반올림한 정수

따라서,  $f(x, y)$ 와  $\hat{f}(x, y)$ 는 다음과 같이 표현된다.

$$f(x, y) = \text{ROUND} [ 0.299 R(x, y) + 0.587 G(x, y) + 0.114 B(x, y) ] \quad (13)$$

$$\hat{f}(x, y) = G(x, y) \quad (14)$$

여기에서,  $R(x, y)$ ,  $G(x, y)$ ,  $B(x, y)$ 는 각각 좌표  $(x, y)$ 의 위치에 있는 화소의 R, G, B 값을 나타낸다.

표 4는 3가지 실험 대상 영상의 PSNR 값을 보여 주고 있다. 표 4의 PSNR 값들을 살펴보면, color 글자가 많은 DOS color 화면이 상대적으로 PSNR이 낮고, 글자의 색깔이 R, G, B 값이 같은 색깔이며(  $R = G = B = 168$  ) 검은 바탕이 대부분을 차지하는 DOS 흑백 화면이 PSNR이  $\infty$ 로 높게 나타나고 있음을 알 수 있다. DOS 흑백 화면의 경우, MSE 값이 0이어서 PSNR이 무한대로 나타나고 있다.

DOS color 화면의 경우, G 값을 그대로 gray level로 사용하여 경계를 검출할 경우, 상대적으로 왜곡이 커서, 수직 선이 경계 검출률이 떨어질 것으로 예상된다.

Table 4. Comparision of PSNR in each sample image

	Window-95	DOS color	DOS B/W
PSNR [dB]	34.82	27.17	$\infty$

세 가지 실험 대상 영상에 대하여, R, G, B 값을 gray scale 변환하여 얻은 gray level을 사용하여 경계를 검출하는 경우와 G 값을 그대로 gray level로 사용하여 경계를 검출하는 경우의, 문자의 수직 천이 경계 검출 성능을 비교한 것이다. G 값만을 사용하여 수직 천이 경계를 검출하는 경우, G 값이 포함되어 있지 않거나 작은 비율을 차지하는 색깔의 문자가 있는 DOS color 화면은, R, G, B 값을 모두 사용하는 경우에 비하여 검출률이 떨어지는 것을 확인할 수 있다. file manager 화면이나 DOS B/W 화면은 G 값만을 사용하는 경우와 R, G, B 값을 사용하는 경우에 모두 100%의 검출률을 보여주고 있다.

## 7. 결 론

멀티미디어화의 추세에 따라 TV, 오디오, PC등의 경계가 점차 모호해지며 최근에 출시되는 PC 제품은 기본적으로 멀티미디어 기능을 제공하고 있다. 이에 반하여 TV 기능을 기본으로 하고 PC와 오디오의 기능을 포함한 멀티미디어 TV는 디지털TV의 전 단계의 형태로 나타나고 있으나 TV는 PC에 비하여 일반 사용자들에게 훨씬 친숙하다는 장점이 있기 때문에 TV의 발전된 형태인 멀티미디어 TV는 멀티미디어 PC에 못지 않은 시장성을 가질 것으로 예상할 수 있다.

멀티미디어 TV의 화질 개선은 TV의 NTSC/PAL 비디오 신호를 발생하기 위해 PC의 VGA 순차 주사 데이터를 비월 주사 형식으로 변환할 때 수반되는 화면의 깜박거림 혹은 경계의 평탄화로 인한 문자의 흐려짐(Smoothing)을 최대한 억제하는 것이다. 이 문제를 해결하는 방법으로 frame memory를 이용하여 이중 주사하는 방법과 영상의 영역별로 주사 변환 방식을 제어하는 방법이 있는데 본 연구에서는 라인 메모리를 사용하는 두 번째의 방법을 채택하였다.

영상의 영역별로 주사 변환 방식을 제어하기 위해서는 먼저 영상에서 문자의 수직 천이 경계 영역을 분리해내는 것이 필요하므로 순차 주사 영상의 2 라인, 3 라인을 이용하여 수직 천이 경계를 검출하는 알고리즘을 제안하였다. software simulation 결과 2 라인과 3 라인을 이용하는 알고리즘 모두 DOS color 영상을 제외하고는 100%의 수직 천이 경계 검출률을 보였고 3 라인을 이용한 알고리즘이 3배 정도 2 라인의 방법보다 빠른 실행시간을 나타내었다. 2 라인을 이용한 실험에서 DOS color 영상은 R, G, B 값을 모두 사용한 경우에 99.94%의 검출률을 보였고 G 값만을 사용한 경우는 98.30%의 검출률을 기록했다. 따라서 G 값만을 사용하고 2 라인을 이용하는 경우에도 납득할만한 수준의 검출률을 얻을 수 있을 것이다.

제안된 경계 검출 알고리즘을 적용하여 PC의 병렬 포트를 통해 VGA 순차 주사 프레임의 라인별로 CH7001 VGA to NTSC/PAL encoder chip를 사용하여 시험용 보드를 제작하였다.



이 보드를 이용한 실험을 통하여 각각의 영역별로 주사 변환 방식을 선택적으로 제어하는 방식이 전체 화면을 0:1:0 모드로 변환하는 방식에 비하여 깜박거림은 줄어들며 문자의 인지도는 그대로 유지되는 결과를 얻었다. 이 실험의 결과로 경계 검출 알고리즘과 제어 루틴을 EPLD에 프로그래밍하여 최종적인 하드웨어를 완성하기 이전에 영역별로 주사 변환 방식을 제어하는 방식의 유용성을 확인 하였다.

본 연구 결과를 통해 영역별로 주사 변환 방식을 선택적으로 제어하는 방식이 깜박거림은 적으면서 문자의 명확성을 그대로 유지 할 수 있음을 확인하였다. 영역별 주사 변환 모드 제어 방식은 이중 주사 방식과 달리 frame memory를 사용하지 않고 line memory 만으로 구현이 가능하므로 본 연구의 결과가 멀티미디어 TV의 화질 개선에 경제적인 해결책으로 활용될 수 있을 것이다.

### 참고문헌

- [1] Keith Jack, *Video Demystified: A Handbook for the Digital Engineer*, HighText Publications, 1993.
- [2] Jun Ohya, Akio Shio, Shigeru Akamatsu, "Recognizing Characters in Scene Images", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, No. 2, February 1994.
- [3] 장문수, "히스토그램과 에지 검출에 의한 문서 영상의 개선된 이진화 알고리즘", 고려대학교 전자공학과 석사논문, 1994.
- [4] Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, Addison-Wesley, 1992.
- [5] Robert J. Schalkoff, *Digital Image Processing and Computer Vision*, John Wiley & Sons, 1989.
- [6] CH7001 Preliminary notes, Chrontel, San Jose, 1995.
- [7] 노웅기, 이승용, "IBM-PC 비디오 카드 프로그래밍", 도서출판 세화, 1993.
- [8] 김태진 역, "고급 프로그래머를 위한 EGA/VGA", 가남사, 1992.
- [9] *Borland C++ Programmer's Guide*, Borland International, 1991
- [10] C. Wayne Brown, Barry J. Shepherd, *Graphics File Formats: Reference and Guide*, Manning Publications, 1995.
- [11] Steve Rimmer, *Bit-Mapped Graphics*, Windcrest Books, 1990.
- [12] 이병배, "C로 PCX 파일 다루기", 에이스 출판사, 1994.