

2D/3D 변환을 위한 Convolution filter

*송혁 **배진우 *최병호 ***유지상

*KETI, **연세대학교 **광운대학교

*hsong@keti.re.kr

Convolution filter for 2D to 3D conversion

*Hyok Song **Jinwoo Bae **Byeongho Choi ***Jisang Yoo

*KETI, **Yonsei Univ. **Kwangwoon Univ.

요약

3DTV는 아나로그 TV 및 HDTV의 차세대 이슈로 부상하였다. 그러나 대부분의 콘텐츠가 2D로 획득되어 저장되어 있으므로 2D 콘텐츠의 3D로의 변화가 필수적이다. MPEG 및 JVT에서 표준화가 진행되고 있으며 이를 위해 국내외 연구소, 학교, 및 업계가 관심을 가지고 참여하고 있다. 2D/3D 변환은 오래전부터 연구되어 왔으나 실제 응용에서는 기대에 못 미치고 있다. 본 논문에서는 FPGA에 기반하고 VHDL로 코딩하여 2D/3D 변환을 위한 Convolution filter를 적용하였다. 좌우 영상을 생성하기 위하여 Convolution filter로 좌우 영상을 왜곡하였다. 필터의 사용으로 사용자의 위치나 취향에 따라서 영상의 왜곡을 달리하여 효과의 변화를 줄 수 있다.

1. 서론

입체감이 있는 사진, 영화를 보고 싶다는 소망으로부터 19세기 중엽의 입체사진을 비롯한 갖가지 입체 표시 원리에 대한 고찰이 유럽에서 시작되었다. 또한 아나로그 TV로부터 시작된 흐름은 HDTV의 상용화에 이르렀고 차세대 영상통신으로 입체감을 제공할 수 있는 3DTV가 주목 받고 있다. 나아가서는 관찰자의 시점에 따라 각각 다른 영상을 보여주는 입체 감각 방송을 의미한다. 오락 및 Animation 산업과 CAD, Simulation 분야 등에서 이러한 파라다임이 시작되고 있다[1]. 일반적으로 입체감을 느낄 수 있는 스테레오 영상을 얻기 위해서는 양안에 해당하는 두 대의 카메라 시스템인 스테레오스코픽 카메라를 이용하여 두 영상이 동시에 획득되는 3D 콘텐츠를 획득하며, 획득된 영상은 압축 처리 과정을 거친 후 전송한다. 3DTV를 위한 큰 흐름으로 MPEG과 JVT의 활동을 들 수 있다. MPEG에서는 3DAV 기술의 실험을 2001년부터 시작하여 3D video와 관련하여 많은 application과 기술들이 제안되었다[2][3]. 유럽에서는 COST230 프로젝트를 시작으로 PANORAMA, ATTEST 및 3DTV 프로젝트로 진행할 정도로 3DTV에 대한 열기가 매우 뜨겁다.

그러나 현재 대부분의 카메라 시스템은 일안 카메라를 이용하여 획득되는 2D 콘텐츠이므로 3DTV에 직접 적용할 수 없다. 이런 문제점을 해소하기 위해서는 기존에 획득/저장되어 있는 2D 콘텐츠를 3D 디스플레이 장치에서 시청할 수 있는 3D 콘텐츠로의 변환이 필요하다.

2D/3D 콘텐츠 변환 기술은 시간차를 가진 연속 영상을 좌우

영상으로 이용하는 방법, 수직시차를 수평시차로 변환하는 방법 및 영상의 Layer를 구분하여 3D로 변환하는 방법 등이 있다 [4]. 다양한 알고리즘의 실제 적용을 위해서는 하드웨어 플랫폼에 기반한 알고리즘의 구현이 필수적이다. 본 논문에서는 FPGA를 이용하여 하드웨어 플랫폼을 구성하였고 이를 실제 3D LCD에 적용하여 실현하였다.

본 논문의 구성은 다음과 같다. 2절에서는 인터폴레이션 필터를 소개하고 3절에서는 필터를 이용한 2D/3D 변환 알고리즘을 설명한다. 4절에서는 하드웨어 플랫폼과 실험 결과를 보이고 5절에서는 결론 및 향후 과제를 언급한다.

2. 인터폴레이션

인터폴레이션의 응용범위는 매우다양하다. 영상의 geometric process에 있어서 Scale, rotation 그리고 Translation이 있다 [5]. 원영상과 처리된 영상사이에서 Hole과 Overlap이 발생하게 된다. Hole은 픽셀이 정해지지 않고 값이 없는 것을 의미하고 Overlap은 그 픽셀을 생성할 때 관여하는 원영상의 픽셀이 2개 이상일 때 발생한다. 인터폴레이션은 Hole과 Overlap이 발생할 때 매우 유용하다. Hole의 경우 인터폴레이션을 주변의 픽셀들을 이용하여 새로운 픽셀값을 생성하게된다. 적절한 인터폴레이션 함수를 선택하는 것은 매우 중요하다. 영상의 화질과 처리 시간은 서로간에 Trade-off가 존재하기 때문이다. 복잡한 알고리즘을 사용하여 영상의 화질을 좋게 만들 수 있는 반면, 알고리즘은 복잡한 함수를 사용하게되어 처리시간이 길어지게 된다.

가장 간단한 방법으로는 Nearest neighbor interpolation이 있다. 이는 새로이 만들어질 픽셀에 가장 가까운 픽셀들중에서 선택하는 방법이다. 이 방법은 원영상에서 하나의 픽셀이 선택되기 때문에 새로운 픽셀을 위하여 연산이 전혀 필요가 없으나 블록화 현상이 발생할 수 있다.

많이 쓰이는 인터폴레이션 방법으로 Bilinear interpolation이 있다. 생성되는 픽셀값은 가중치가 주어진 주변 픽셀들의 합으로 만들어진다. 가중치는 Linear 하게 주어지며 Liemar한 가중치는 생성될 픽셀에서의 원근치에 따라 주어진다. weight를 이용하여 생성된 픽셀과 P3 및 P4에 의해 생성된 픽셀을 이용하여 원하는 위치의 새로운 픽셀을 생성할 수 있다.

Higher order interpolation은 Nearest neighbor interpolation 및 Bilinear interpolation보다 더 확장된 개념으로 cubic convolution 방식과 B-spline 방식이 많이 쓰인다[6][7]. 그림 xx와 같이 구하려는 픽셀의 중간에 함수가 위치하게 되며 원영상의 픽셀값의 곱의 합으로 구해지게 된다.

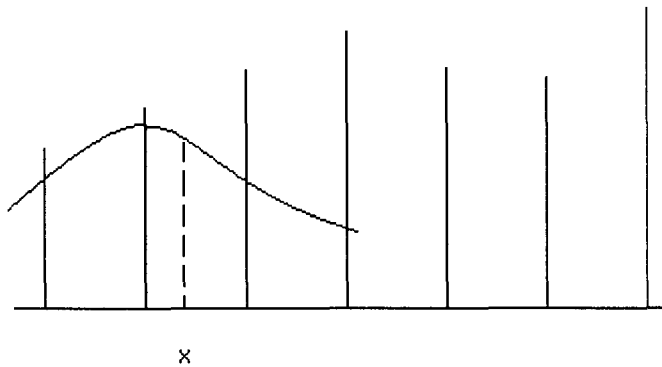


그림 xx

그림 xx의 인터폴레이션 함수는 아래와 같이 정의할 수 있다.

$$\hat{f}(x) = \sum_k c_k \beta(x - x_k)$$

여기에서 $\hat{f}(x)$ 는 인터폴레이션 함수이고 $\beta(x)$ 는 인터폴레이션 커널이다. 그리고 x 와 x_k 는 각각 연속함수의 값과 discrete 값을 의미한다.

본 논문에서는 Cubic convolution filter를 사용하였다. Cubic convolution filter는 filter의 파형이 양 끝에서 고정된 값을 가지므로 연산이 용이하다. 아래는 일반적인 Cubic convolution 함수이다.

$$\begin{aligned} \hat{f}(x) = & \alpha_k (f(x_{k-1}) - f(x_{k+1}))A(s) \\ & + \alpha_k (f(x_k) - f(x_{k+2}))B(s) \\ & + (f(x+k) - f(x_{k+1}))C(s) \\ & + f(x_k) \end{aligned} \quad \text{식(1)}$$

여기에서

$$\begin{aligned} A(s) &= (s^3 - 2s^2 + s) \\ B(s) &= (s^3 - s^2) \\ C(s) &= (2s^3 - 3s^2) \end{aligned} \quad \text{식(2)}$$

이다.

3. 2D/3D 변환

현재 대부분의 콘텐츠는 2D로 획득/저장/압축된다 입체로 표현되는 스테레오 스코프 영상은 두 개의 다른 각도의 영상으로 표현된다. 현재는 스테레오스코프 카메라가 개발되어 3D 데이터로도 저장되기도 하나 소수이므로 2D/3D 변환이 필요하다.

Garcia 의한 시각적 특성을 이용하여 시간 차이를 둔 연속 영상을 이용하여 좌우 영상을 구성하면 3D 효과를 만들어 낼 수 있다[8]. 이 방법은 매우 간단하면서도 좋은 효과를 만들어 낼 수 있어 쉽게 응용 가능하다. 연속된 영상의 특성을 파악하여 시간 차이를 조절하여 적절한 좌우 영상은 선택하여 깊이감을 조절하여 시청자에게 피로감을 주지 않은 시차를 만들어낸다.

영상의 일반적인 특성을 이용하여 3D 효과를 만들어내는 방식으로 영상의 상하에 배치된 객체와 배경에 시차를 달리 부여하는 방식이 있다. 일반적인 영상은 상단은 영상의 배경으로 객체에 비해 먼 위치에 있고 객체는 아래쪽에 가까운 쪽에 배치되는 특성을 이용하여 시차를 달리 만들어 낼 수 있다. 이러한 효과는 필터를 이용하여 영상을 왜곡하여 만들어낸다. 본 논문에서는 Cubic convolution filter를 이용하여 영상을 왜곡하였다.

그림 xx에서 $\hat{f}(x)$ 의 함수를 다시 샘플링할 때에 샘플링하는 위치를 조절하기 위하여 식(1)과 식(2)를 이용하여 filter를 변형하여 아래와 같은 식을 얻을 수 있다. 식(1)은 아래 식과 같이 쓸 수 있다.

$$\begin{aligned} \hat{f}(x) = & f(x_{k-1})\{\alpha_k(s^3 - 2s^2 + s)\} \\ & + f(x_k)\{\alpha_k(s^3 - s^2) + (2s^3 - 3s^2 + 1)\} \\ & + f(x_{k+1})\{\alpha_k(-s^3 + 2s^2 - s) + (-2s^3 + 3s^2)\} \\ & + f(x_{k+2})\{\alpha_k(-s^3 + s^2)\} \end{aligned} \quad \text{식(3)}$$

여기에서 s 를 구하기 위한 식은 아래와 같다.

$$\begin{aligned} f_{temp} &= B_{hor} \times i - \frac{j}{H} \times D - UD \\ J_{int} &= (int) f_{temp} \\ s &= f_{temp} \times J_{int} \end{aligned} \quad \text{식(4)}$$

B_{hor} 는 입력사이즈와 출력사이즈의 비율이고 i 는 픽셀의 수평 위치이다. j 는 픽셀의 수직 위치이고 H 는 영상의 높이이다. D 는 왜곡하는 영상의 중심 기준을 의미하며 UD 는 왜곡되는 전체 영상의 시차를 의미한다. 식(3)을 이용하여 영상은 그림 xxx와 같이 다양한 모양의 왜곡이 가능하다. 그림 xxx에서 짙은 영역은 원영상이 되고 옅은 영역은 왜곡된 영상을 나타낸다.

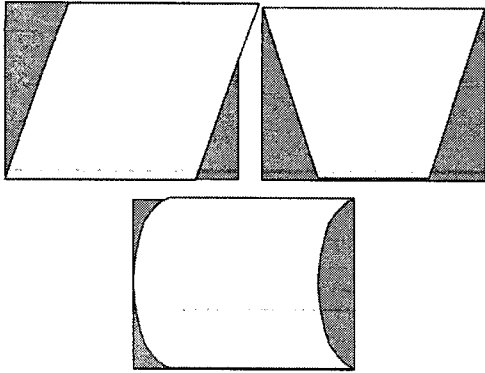


그림 xxx. 필터를 이용한 영상의 왜곡

식(3)를 이용하여 영상의 특성을 이용하여 상하 시차를 달리 하여 그림 o과 같은 결과를 얻었다. 그림 o는 PC 시스템에서 소프트웨어적인 결과이다.

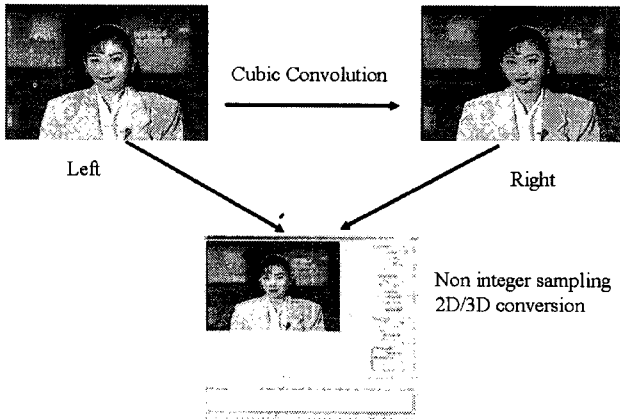


그림 o. 상하 시차를 달리한 영상

3D LCD에서는 그림oo와 같이 좌우 영상을 혼합하여 출력하면 3D LCD의 종류에 따라 다른 방식으로 표현하게 된다. 그림 xxx는 Field sequence 방식의 혼합방식이다.

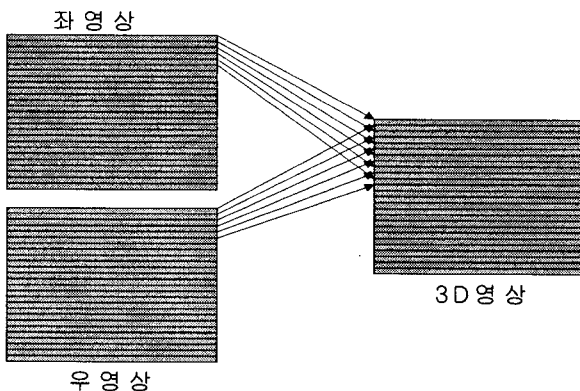


그림 oo. Field sequence 방식의 3D display

3D로 표현하는 방식은 주로 Field sequence, Frame sequence, side by side 등이 많이 쓰인다. Field sequence 방식의 장점은 좌우 영상이 모두 일반적인 영상의 주파수와 같은 주파수이므로 시청자에게 피로감을 주지 않는다. Field sequence 방식을 위하여 VHDL이 코딩된 FPGA 보드는 4개의 영상 프레

임이 저장되는 공간을 구성하였다. 입력되는 영상은 메모리에 순차적으로 저장되며 하나의 프레임이 저장되면 연속되는 프레임은 다른 메모리에 저장을 하였다. 저장된 영상을 왜곡하여 그림 oo와 같이 Field sequence 방식으로 혼합하여 LCD에 디스플레이하였다.

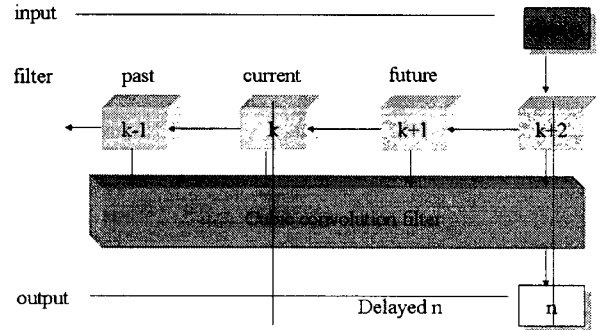


그림 000.

수식(3)과 같이 왜곡된 하나의 픽셀을 연산하기 위해서는 네 개의 픽셀이 필요하다. 즉 출력되는 영상은 그림 000과 같이 가장 늦게 입력되는 픽셀과 같이 출력된다. 이를 LCD에 적용하기 위해서는 Horizontal sync 신호의 조작이 필요하다.

4. 실험결과

본 실험은 600,000개의 게이트를 가진 Actel APA600-BG456을 이용하여 하드웨어를 구성하였다. 사용된 소프트웨어는 Modeltech사의 Modelsim을 이용하였다.

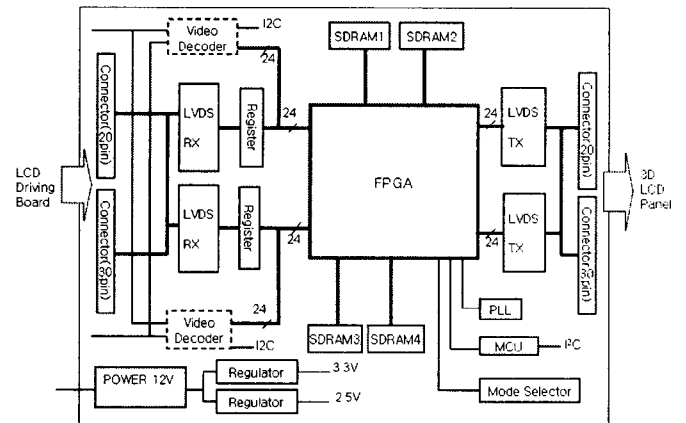


그림 0000. 제작된 FPGA 보드의 블록도

그림 0000는 그림 0x에 보이는 바와 같이 제작된 보드의 블록도이다. LCD 신호를 FPGA를 이용하여 영상 데이터를 메모리에 저장한 후에 저장된 영상 데이터를 Convolution filter를 이용하여 왜곡한 후에 출력한다. 모든 프로세스는 실시간으로 동작하여 실제 LCD에 적용하여 장시간 동작에도 무리 없이 동작하였다.

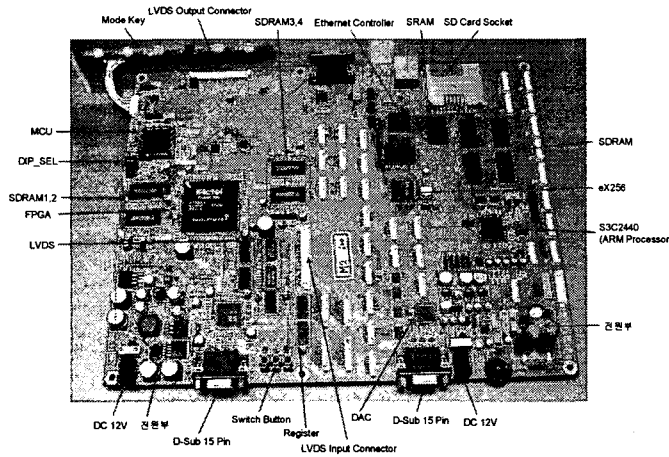


그림 0x. 제작된 플랫폼 보드

5. 결론 및 향후연구과제

FPGA를 이용하여 실제 3D LCD에 적용하여 구성하여 좋은 결과를 보였다. 본 결과는 LCD 뿐 아니라 모든 디스플레이 장치에 적용 가능하여 다양한 응용 분야에 쓰일 수 있다. Convolution filter를 이용한 2D/3D 변환은 영상 전체에 왜곡을 주어 3D 효과를 만들어 내었다. 이 방식은 영상내에서 객체의 움직임 및 카메라의 움직임에 따라 3D 효과가 달라지게 된다. 이러한 취약점을 보완하기 위해서는 Optical flow를 이용한 정확한 움직임 유형 파악, 객체 추출 및 객체별 시차 구성을 하여야 하며 응용 분야에 따라서 Virtual reality를 응용하여 Mesh 코딩등을 이용하여 별도로 코딩하는 연구가 계속 되어야 한다.

[참고문헌]

- [1] 2001 신기술동향조사 보고서 3차원 입체영상기술, 특허청 2001
- [2] ISO/IEC JTC1/SC29/WG11, "Call for Comments on 3DAV", Doc. N6051, Gold Coast, Australia, October 2003
- [3] T. Okino, et. al, "New television with 2D/3D image conversion technique", SPIE Vol. 2653, Photonic West, 1995
- [4] B.J.Garcia, "Approaches to stereoscopic video based on spatial-temporal interpolation", SPIE Vol. 2635, Photonic West, 1990
- [5] Randy Crane, "A simplified approach to image processing", Prentice hall, 1997
- [6] Michael Unser, Akram Aldroubi, Murray Eden, "Fast B-spline Transforms for Continuous Image Representation and Interpolation", IEEE Transactions on pattern analysis and machine intelligence Vol. 13, 1991
- [7] Jong-ki Han, Seung-ung Baek, "Parametric cubic convolution scaler for enlargement and reductio of image", IEEE transactions on Consumer Electronics, Vol. 46, 2000
- [8] T. Okino, "New television with 2D/3D image conversion