

# 색역폭 매핑을 이용한 디지털 TV 디스플레이 장치의 화질 개선

한동일

세종대학교 컴퓨터공학과

## Digital TV Display Quality Enhancement Method Based on the Color Gamut Mapping

Dongil Han

Department of Computer Engineering, Sejong University

E-mail : dihan@sejong.ac.kr

### Abstract

본 논문에서는 새로운 색역폭(color gamut) 매핑 방법을 이용하여 디지털 TV 디스플레이 장치의 화질을 개선하는 방법을 제안하였다. 기존에 실시간 적용이 어렵던 색역폭 매핑 방법을 실시간으로 처리하기 위한 하드웨어 구조를 제안하였으며 이를 통하여 수 나노 초 단위의 처리 속도가 필요한 디지털 TV의 디스플레이 장치에 성공적으로 적용이 가능하였다. 또한 제안된 하드웨어 구조는 필요에 따라 색역폭 매핑 해상도의 조절이 가능하여 해상도 및 하드웨어 구현 비용을 적절히 조절할 수 있는 장점이 있다.

### I. 서론

최근 디지털 TV 방송이 시작됨에 따라서 고해상도의 디스플레이 장치를 장착한 디지털 TV의 보급이 확대되고 있다. 그리고 기존의 아날로그 TV의 대다수를 차지하고 있던 CRT 디스플레이 장치의 경우 대형화가 어려운 단점으로 인해서 디지털 TV 시대에서는 점차 사용 비율이 줄어들 것으로 예측되는 반면 PDP TV, LCD TV, LCD 프로젝션 TV, DLP 프로젝션 TV 등 대화면이 용이한 디스플레이 장치들의 사용이 점차 증가될 것으로 예측되고 있다. 그러나 새롭게 개발되는 디스플레이 장치들의 경우 대화면이 용이한 반면 CRT 디스플레이 장치와 비교하여 화질, 시야각, 명암비, 밝기, 수명 등 여러 가지 면에서 CRT의 성능에 못 미치고 있는 것도 사실이다. 특히 제일 중요한 화질의 경우 CRT 디스플레이 장치의 화질과 비교하면 그 수준이 떨어지고 있는 실정이며 영상 처리 기법을 이용한 화질 개선과

관련하여 다방면의 연구 개발이 이루어지고 있다[1~2].

특히 본 논문에서는 기존에 PC용 칼라 모니터와 칼라 프린터 간의 색 재현성의 차이를 보정하는 데에 사용하는 색역폭 매핑(color gamut mapping)의 개념[3~6]을 디지털 TV 분야에 적용하여 디스플레이 장치의 색 재현성을 개선하는 방법을 제안하고자 한다.

공간 영역의 색역폭 매핑 기법[3]은 밝기의 변화를 일정하게 유지시키면서 각 화소별 색역폭 매핑 방법의 단점을 극복하기 위하여 제안된 방법이며 좋은 실험 결과를 제공한다. 또 다른 접근 방법으로 다양한 앵커 점들에 대한 밝기 정보 매핑[4] 방법이 있다. 밝기 정보 매핑을 통하여 두 색역폭 간의 최대의 크로마 값을 갖는 위치에서의 밝기 값의 차이를 최소화하는 기법이다. 또 다른 접근 방법으로서 3차원의 색역폭 축소 기법[5]에서 제안되었다. 원 영상과 프린터 장치간의 색역폭의 관계를 고려한 후  $L^*a^*b^*$  공간에서 색역폭을 축소하는 방법을 사용하고 있으며 기존의 이차원의 색역폭 매핑 기법에 비해서 우수한 결과를 제공한다.

이 밖에 다양한 색역폭 매핑 기법이 제안되고 우수한 성능을 제공하고 있지만 대부분의 색역폭 매핑 기법이 디스플레이 장치에 실시간으로 적용되기에는 매우 큰 어려움이 존재한다. 색역폭 매핑 알고리즘이 디지털 TV의 디스플레이 장치에 적용이 되기 위해서는 약 10나노 초 정도의 처리 속도가 필요하며 기존의 알고리즘을 직접 이용해서 구현하기는 거의 불가능하다.

따라서 본 논문에서는 해상도 질감 3-차원 특업 테이블의 개념을 이용하여 색역폭 매핑을 실시간으로 구현할 수 있는 하드웨어 구조를 제안하고 이를 통해서

기존의 다양한 색역폭 매핑 알고리즘을 적용하여 디지털 TV의 화질을 실시간으로 개선할 수 있는 색역폭 매핑 방법을 제안한다.

## II. 해상도 절감 3차원 룩업 테이블

균일 색 공간으로서 인간의 시각 특성을 가장 잘 표현하는 색 공간으로서 CIE  $L^*a^*b^*$  색 공간을 들 수 있다. 비록  $L^*a^*b^*$  색 공간이 직접 디스플레이 용도로는 사용되지 않으나 색을 표현하는 장치인 칼라 모니터, 칼라 스캐너, 칼라 프린터 등의 색의 표현 특성을 정확하게 표현하는 장점을 가지고 있다. 그러나 일반적인 디스플레이 장치들은 RGB 색 공간을 사용하고 있다. RGB 색 공간의 경우 3 가지 종류의 원색, 즉 Red, Green, Blue 의 삼원색을 이용하여 삼원색의 선형 조합으로서 다양한 색을 만들어 내게 된다. 그래서 대부분의 색역폭 매핑 알고리즘들은 다음 그림 1 과 같은 구조를 이용하여 색역폭 매핑을 구현하고 있다.

이러한 색역폭 매핑 방법을 하드웨어로 구현하기 위해서 두 개의 색좌표 변환기와 색역폭 매핑부를 직접 하드웨어로 구현하는 방법을 생각할 수 있다. 그러나 RGB 색 공간과  $L^*a^*b^*$  색 공간의 변환식이 매우 비선형적이기 때문에 이를 실제로 하드웨어로 구현하기는 매우 어렵다. 그리고 색역폭 매핑부 또한 하드웨어로 구현하기에 매우 어려운 특성이 있다. 또 다른 방법으로는 두 개의 색좌표 변환기능과 색역폭 매핑 기능을 테이블로 만들어서 3-차원 룩업 테이블로 구성하는 방법을 들 수 있다. 그러나 이 경우 필요한 기억 장치의 양은  $256 \times 256 \times 256 \times 3 \text{ byte} = 50,331,648 \text{ byte}$  로서 하드웨어로 구현하기에는 거의 불가능한 용량이 필요하게 된다.

따라서 본 논문에서는 3-차원 룩업 테이블의 해상도를 낮추고 나머지 부분은 보간을 통하여 계산해 냄으로써 삼차원 룩업 테이블과 거의 동일한 성능을 제공하는 해상도 절감 삼차원 룩업 테이블(Reduced Resolution 3-D Look-up Table) 구조를 제안하고 이를 이용

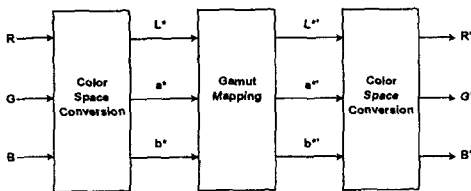


그림 1 전형적인 색역폭 매핑 단계

하여 필요한 하드웨어의 양을 대폭 줄이면서 실시간으로 색역폭 매핑을 구현 가능한 방법을 제안한다.

그림 2 에 해상도 절감 3-차원 룩업 테이블의 개념도를 나타내었다. 여기에서는 R, G, B 각각 256 계조를 갖는 색 성분의 MSB 3 비트 정보를 이용하여 룩업 테이블을 구성한다. 그림 3 에 나타낸 바와 같이 해상도 절감 룩업 테이블을 이용하여 색역폭 매핑이 가능하기 위해서는 R, G, B 각각 MSB 3 비트 정보를 이용하여 각각의 구간을 9 개의 구간으로 나눈 뒤 매핑이 이루어져야 할 화소를 포함하는 입방체에서의 색역폭 매핑 값을 제공해 줄 수 있어야 한다. 그러므로  $9 \times 9 \times 9$  크기의 3-D 룩업 테이블에서는 주어진 R, G, B 3 비트 값을 입력 받아서 해당 화소를 둘러싸는 8 개의 좌표 점에 대해서 새로운 색역폭 매핑 값을 제공해 주어야 하며 이때 R, G, B 각각에 대해서 새로운 매핑 값을 동시에 제공해 주어야 한다.

그리고 나머지 LSB 5 비트 값을 이용하여 최종 색역폭 값을 보간을 통하여 구성하게 되는데 그림 4 에 3-차원 보간 기능을 자세히 나타내었다.  $9 \times 9 \times 9$  3-차원 룩업

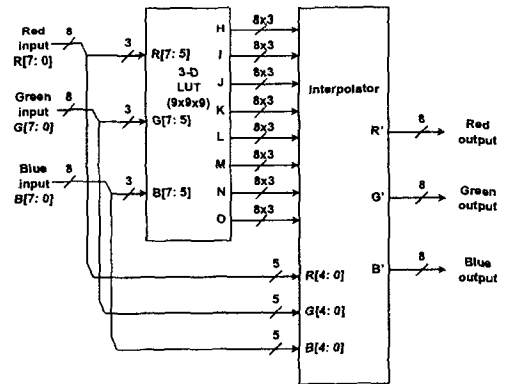


그림 2 해상도 절감 삼차원 룩업 테이블 구조

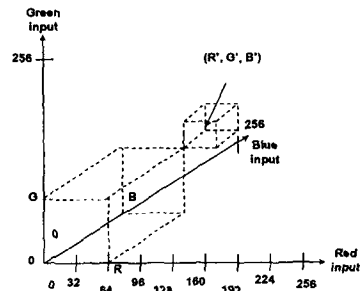


그림 3 해상도 절감시의 매핑 개념

테이블에서 8 개의 정육면체 꼭지점에 해당하는 색역폭 매핑 값을 R, G, B 각각에 대해서 제공하게 되며 이 값과 R, G, B 각각의 LSB 5 비트 값을 이용하여 해당 위치에서의 색역폭 매핑 값을 보간을 이용해서 계산해 낼 수 있다. 그림 4 에서는 처음에 H 점과 I 점에서의 R, G, B 특업 테이블 값과 R[4:0] 값을 이용하여 HI 점에서의 색역폭 매핑 값을 계산할 수 있으며 마찬가지로 J 점과 K 점에 대해서도 R[4:0] 값을 이용하여 JK 점에서의 매핑 값을 계산해 낼 수 있다. 그 이후 HI 점과 JK 점에서의 매핑 값과 B[4:0] 값을 이용하여 HIJK 점에서의 매핑 값을 계산해 낼 수 있다.

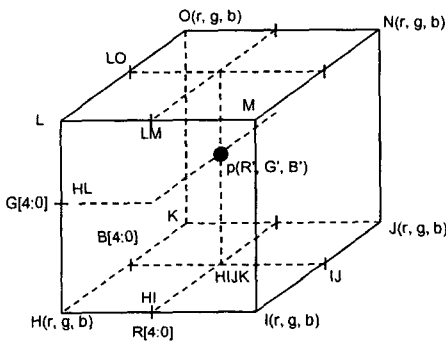


그림 4 보간을 이용한 색역폭 매핑 단계

이와 같은 방법으로 최종적으로 p 점에서의 새로운 매핑 값을 R, G, B 성분 각각에 대해서 계산해 낼 수 있으며 이런 작업을 매 화소마다 반복적으로 수행함으로써 인가되는 RGB 화소 데이터에 대해서 색역폭 매핑된 화소 값 R'G'B' 값을 매 화소마다 계산해 낼 수 있게 된다.

### III. 하드웨어 구현

그림 2 의 3-차원 특업 테이블을 실제 하드웨어로 구현하는 것은 또 다른 문제를 야기한다. 주어진 R, G, B 값에 의해서 한 점의 색역폭 매핑 값을 계산하기 위해서는 동시에 8 개의 좌표에서의 색역폭 매핑 값을 추출해 내어야 하는 문제가 존재한다. 이를 파이프 라인 구조로 변경하더라도 동시에 인접 화소의 매핑 값을 제공할 수 있어야 하기 때문에 같은 문제를 야기하게 된다. 이와 같은 문제를 해결하기 위해선 그림 2 의 9x9x9 3-차원 특업 테이블을 실제적으로는 그림 5 와 같이 8 개의 1 차원 특업 테이블과 어드레스 디코더, 그리고 데이터 스위치로 구성하면 된다.

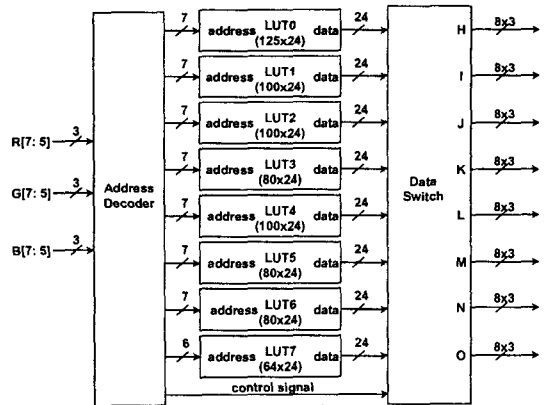


그림 5 9x9x9 특업 테이블의 실제 구현

즉 9x9x9 = 729 개의 좌표점들은 각각 8 개의 1 차원 특업 테이블로 나뉘어서 저장이 된다. 예를 들면 R=0, G=0, B=0 점에 대한 매핑 값은 LUT0 의 첫번째 어드레스에 저장이 되며 R=1, G=0, B=0 점에 대한 매핑 값은 LUT1 의 첫번째 어드레스에 저장이 되며 원점을 포함하는 입방체의 각 점이 LUT0 ~ LUT7 의 첫번째 어드레스에 골고루 저장이 된다. 그리고 색역폭 매핑 과정에서 원점을 포함하는 입방체 내의 한 점이 선택될 때에는 여덟 개의 1 차원 특업 테이블의 첫번째 어드레스 데이터를 출력하여 필요한 보간을 수행하게 된다. 이러한 과정이 중복 없이 수행되기 위해서는 복잡한 어드레스 디코딩 과정이 필요하게 되는데 그 어드레싱 로직을 정리하면 다음과 같이 표현할 수 있다.

$$\begin{aligned} \text{LUT0\_address} &\leq (R+1)/2 + 5*((G+1)/2) + 25*((B+1)/2); \\ \text{LUT1\_address} &\leq (R) /2 + 4*((G+1)/2) + 20*((B+1)/2); \\ \text{LUT2\_address} &\leq (R+1)/2 + 5*((G) /2) + 20*((B+1)/2); \\ \text{LUT3\_address} &\leq (R) /2 + 4*((G) /2) + 16*((B+1)/2); \\ \text{LUT4\_address} &\leq (R+1)/2 + 5*((G+1)/2) + 25*((B) /2); \\ \text{LUT5\_address} &\leq (R) /2 + 4*((G+1)/2) + 20*((B) /2); \\ \text{LUT6\_address} &\leq (R+1)/2 + 5*((G) /2) + 20*((B) /2); \\ \text{LUT7\_address} &\leq (R) /2 + 4*((G) /2) + 16*((B) /2); \end{aligned}$$

RGB 의 특정 MSB 값에 따라서 각 1 차원 특업 테이블에서 제공하는 데이터의 위치는 그림 4 의 좌표점의 임의의 값이 될 수 있다. 즉 LUT0 에서 제공하는 데이터는 RGB 입력 값에 따라서 H 점이 될 수도 있고 L 점이 될 수도 있으며 8 개의 좌표 중 임의의 점이 될 수 있다. 이러한 문제를 해결하기 위하여 데이터 스위치부에서는 입력되는 매핑 값을 적절히 매핑하여 보간부에서는 항상 그림 4 와 같이 구성을 갖도록 색역폭

데이터를 스위칭 하는 역할을 수행한다. 본 논문에서 제안한 하드웨어 구조와 [6]의 색역폭 매핑 알고리즘을 이용하여 색역폭 매핑한 결과를 그림 6 에 나타내었으며 잔디 부분의 색이나 하늘 색이 좀더 깨끗하게 재현됨을 확인할 수 있다.

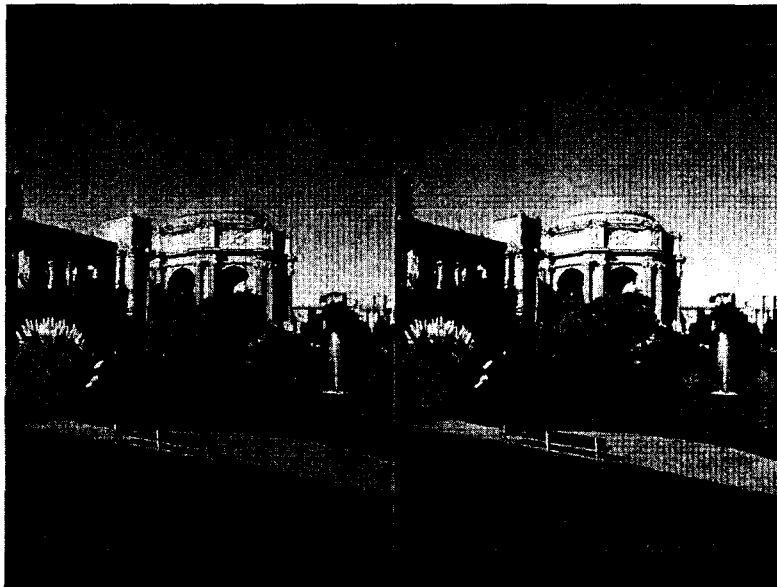
#### IV. 결론

본 논문에서 제안한 하드웨어 구조는 색역폭 매핑 알고리즘의 기법의 종류에 무관하게 실시간으로 적용할 수 있는 장점이 있다. 또한 제안된 하드웨어 구조는 가능한 하드웨어 크기에 따라서 해상도를 조절할 수 있으며 하드웨어나 ASIC 으로 쉽게 구현이 가능한 구조이다. 제안된 하드웨어 구조는 추후 색역폭 매핑 뿐만 아니라 색좌표 변환, 감마 변환 등 기존의 색 처리 알고리즘을 동시에 구현이 가능하여 다양한 색 신호 처리 알고리즘에 범용으로 적용이 가능한 장점이 있다.

#### 참고문헌

[1] D. Han, C.-Y. Shin, S.-J. Choi, and J.-S. Park, "A Motion Adaptive 3-D De-interlacing Algorithm based on the Brightness Profile Pattern Difference," *IEEE Trans. on*

*Consumer Electronics*, vol. 45, no. 3, pp. 690-697, 1999  
 [2] S.-Y. Kim, D. Han, S.-J. Choi and J.-S. Park, "Image Contrast Enhancement Based on the Piecewise-Linear Approximation of CDF," *IEEE Trans. on Consumer Electronics*, Vol. 45, no. 3, pp.828-834, August 1999.  
 [3] R. Bala, R. deQueiroz, R. Eschach and W. Wu, "Gamut Mapping to Preserve Spatial Luminance Variations," *Journal of IS & T*, Vol. 45, no. 5, pp.436-443, 2001.  
 [4] C.-S. Lee, Y.-W. Park, S.-J. Cho and Y.-H. Ha, "Gamut Mapping Algorithm Using Lightness Mapping and Multiple Anchor Points for Linear Tone and Maximum Chroma Reproduction," *J. of IS & T*, Vol. 45, no. 3, pp.209-223, 2001.  
 [5] H. Chen , H. Kotera, "3-dimensional Gamut Mapping Method Based on the Concept of Image Dependence," *J. of IS & T*, Vol. 46, no. 1, pp44-52, 2002.  
 [6] B. H. Kang, M. S. Cho, J. Morovic and M. R. Luo, "Gamut Compression Algorithm Development on the Basis of Observer Experimental Data," in *Proc. 8<sup>th</sup> IS&T/SID Color Imaging Conf.*, IS&T, Springfield, VA 1999, pp. 268-272



(a) 원영상

(b) 색역폭 매핑된 영상

그림 6 색역폭 매핑 결과