

# Cubic Interpolation 을 이용한 Image Upscaler 의 LUT 구성 방법

한재영\*, 이성원\*\*  
 광운대학교 컴퓨터공학과  
 e-mail : \* boui@kw.ac.kr  
 \*\*swlee@kw.ac.kr

## LUT Constructing Method for Image Upscaler using Cubic Interpolation

Jae-Young Han\*, Seong-Won Lee\*\*  
 Dept. of Computer Engineering, Kwangwoon University

### 요 약

영상 확대 기술을 담당하는 업스케일러를 설계 할 때 보간 기법의 선택은 매우 중요하다. 적용하는 보간 기법에 따라 출력되는 영상의 화질과 스케일러 내부의 연산량 및 하드웨어 복잡도가 결정되기 때문이다. 디지털 영상 처리 기술의 발달과 함께 성능이 좋은 여러 가지 보간 기법들이 개발되어 오고 있지만 알고리즘의 복잡도를 고려하지 않은 기법들은 실시간 처리와 하드웨어 적용에 부적합한 경우가 많다. 비교적 좋은 성능을 보여주는 Cubic interpolation 역시 인접 화소 보간 기법이나 선형 보간 기법과 비교하면 훨씬 더 많은 연산량을 요구한다. 이런 높은 연산량의 대부분은 픽셀의 밝기값에 곱해지는 계수를 구하기 위한 복잡한 계산에서 기인한다. 따라서 본 논문에서는 cubic interpolation 의 복잡도를 낮춰 하드웨어 적용에 적합하도록 하기 위하여 LUT(Look-up Table)을 이용하는 방법을 제안하고 실험을 통해 그 결과를 보인다.

### 1. 서론

최근 디지털 영상 기술의 발달로 디스플레이 장치를 갖는 많은 기기들이 출시되고 있다. 이런 기기들은 간편하게 들고 다닐 수 있는 포터블(portable) 기기에서부터 대형 TV 에 이르기까지 그 종류가 매우 다양하며 장착되어 있는 디스플레이 장치의 종류 또한 다양하다. 또한 특정 기기에서 촬영하거나 기기에 저장되어 있는 영상 정보를 별도의 처리 없이 다른 기기에서 재생하고자 하는 소비자들의 욕구가 증대됨에 따라 이종기기 간의 호환성은 점점 더 중요해 지고 있다.

호환성을 높이는 장치 중 하나인 스케일러는 입력 영상을 재생할 기기의 디스플레이 장치에 적합하도록 해상도를 조정해 주는 기능을 수행한다. 따라서 해상도 변환 후 발생하는 화질 열화를 최소화하면서 모바일 기기에도 내장 될 수 있도록 크기와 전력소모가 작은 스케일러를 개발하는 것이 중요하다.

스케일러의 핵심기술에서 보간 기법으로 적용하는 보간 기법에 따라 출력 영상의 화질뿐만 아니라 전체 하드웨어 크기가 좌우된다. 본 논문에서는 평탄한 영역과 경계영역 모두에서 비교적 우수한 성능을 보이는 cubic interpolation 을 스케일러에 적용할 때 하드웨어 복잡도를 낮추기 위하여 LUT 를 구성하고 이용하는 방법을 제안한다.

### 2. Cubic Interpolation 과 LUT 의 구성

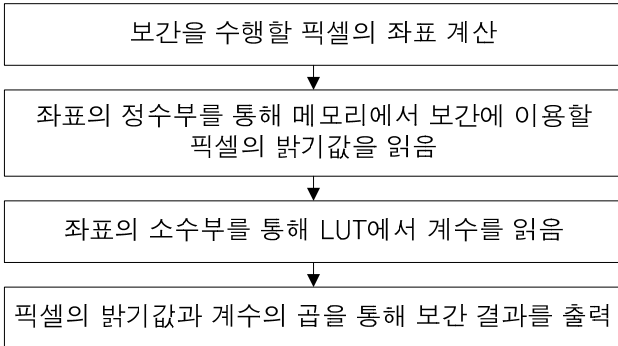
Cubic interpolation 은 경계영역에서의 열화가 적어 고전적이면서도 현재까지 많이 사용되고 있는 보간 기법이다. Cubic interpolation 수식을 점과 그 점에 곱해지는 계수에 대해서 정리하면 식 (1)과 같다[1].

$$f(x) = \left[-\frac{1}{2}s^2 + s^2 - \frac{1}{2}s\right]f(x_{n-1}) + \left[\frac{8}{2}s^2 - \frac{5}{2}s^2 + 1\right]f(x_n) \\ + \left[-\frac{8}{2}s^2 + 2s^2 + \frac{1}{2}s\right]f(x_{n+1}) + \left[\frac{1}{2}s^2 - \frac{1}{2}s^2\right]f(x_{n+2}) \quad (1)$$

식 (1)로부터 cubic interpolation 의 연산량과 복잡도를 높이는 상당부분이 계수 계산에서 비롯됨을 알 수 있다. 따라서 계수 계산에 대한 부분을 미리 계산하여 LUT 에 저장해 놓으면 복잡한 연산을 테이블에 대한 한번의 읽기 연산으로 대체할 수 있어 연산량과 하드웨어 복잡도를 모두 낮출 수 있다.

LUT 를 적용한 스케일러에서는 보간을 위해 그림 1 과 같은 과정을 거친다. 먼저 입력 영상과 출력 영상의 해상도를 알면 출력 영상의 픽셀 간 거리를 알 수 있고, 이를 이용하여 입력 영상 위에서의 출력 영상의 픽셀들의 좌표를 계산할 수 있다[3]. 계산된 좌표의 정수 부는 보간에 이용할 입력 영상의 픽셀 좌표이므로 메모리로부터 해당 위치의 픽셀 정보를 읽어오기 위해 사용되며, 좌표의 소수 부는 수식 (1)의

변수  $s$  에 해당하며 계수 값을 결정하는 요소이므로 LUT로부터 계수를 읽어 오기 위해 사용된다. 최종적으로 읽어온 픽셀의 정보와 계수 간의 간단한 연산으로 결과를 얻을 수 있다.



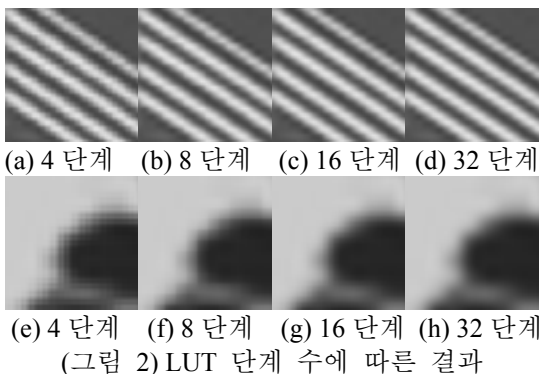
(그림 1) 보간이 수행되는 순서도

LUT로부터 계수를 읽어오는 과정에서 LUT의 크기는 유한하므로 좌표의 소수부를 LUT의 특정 주소로 맵핑(mapping)하는 과정이 필요하다. 즉, 좌표의 소수부는 유한개의 단계로 양자화되며, 양자화시키는 단계의 수에 따라 LUT의 크기와 영상의 화질이 달라지게 된다. 단계의 수를 늘릴수록 화질은 좋아지지만 LUT의 크기가 커져 전체 하드웨어 크기 또한 커지므로 LUT의 크기와 결과 영상의 화질 간의 적절한 타협점이 필요하다.

### 3. 실험 결과

그림 2는 LUT의 단계 수에 따른 결과 영상으로 a, b, c, d는 SVGA 영상을 QXGA 영상으로 확대하였을 때를 보여주며 e, f, g, h는 QVGA 영상을 QXGA 영상으로 확대하였을 때의 결과를 보여준다. 표 1은 그림 2에 대한 PSNR을 보여준다.

(QVGA: 320x240, SVGA: 800x600, QXGA: 2048x1536)



(그림 2) LUT 단계 수에 따른 결과

그림 2의 결과영상으로부터 비교적 영상 확대 비율이 작은 경우에는 LUT의 단계를 8 단계 이상으로 나눌 때에는 주관적인 화질의 차이가 크지 않음을 알 수 있으며, 비교적 영상 확대 비율이 큰 경우에는 16 단계 이상으로 나눌 때 차이가 크지 않음을 알 수 있다.

<표 1> LUT 단계 수에 따른 PSNR

	4 단계	8 단계	16 단계	32 단계	64 단계
a,b,c,d	29.76	29.59	29.62	29.65	29.69
e,f,g,h	25.21	25.21	25.24	25.32	25.32

표 1을 보면 SVGA를 확대했을 때 4 단계에서 PSNR 값이 가장 높게 나오지만 8 단계부터 안정적으로 나오는 것을 확인할 수 있으며 QVGA를 확대했을 때는 32 단계부터 안정적으로 높은 PSNR 값이 나옴을 확인할 수 있다. 따라서 여러 종류의 해상도를 고려했을 때 LUT의 크기는 32 단계 또는 64 단계 정도가 적당하다고 할 수 있다.

### 4. 결론

보간 시 필요한 네 개의 계수를 표현하기 위해 사용되는 bit 수를 모두 합쳐 n-bits 라고 한다면 LUT를 32 단계로 나누었을 때 LUT의 크기는 4nBytes(32\*n-bit)가 되며, 64 단계로 나누었을 때는 8nBytes가 된다. LUT를 사용하지 않고 덧셈기와 곱셈기를 이용하여 스케일러를 구현하면 네 개의 계수를 계산하기 위해서 최소 2 개의 곱셈기와 11 개의 덧셈기가 필요하다. 실제로 2 차원 평면 위에 있는 영상을 보간하기 위해서는 최소 두 배의 자원인 4 개의 곱셈기와 22 개의 덧셈기가 필요하며 64 단계로 나누어진 LUT를 사용하면 16nBytes 크기의 메모리로 대체 할 수 있다. 보다 정확한 크기 비교를 위해서는 계수 값의 정확도와 관련된 n 값을 결정하기 위한 연구가 필요하다.

LUT를 이용하면 많은 덧셈과 곱셈 대신 메모리에 대한 한번의 읽기 연산만으로 계수 값을 알 수 있기 때문에 하드웨어의 크기 측면뿐만 아니라 연산 속도 측면에서도 큰 이점을 갖고 있다. 따라서 LUT를 이용하는 기법은 실시간으로 처리하는 스케일러를 구현하기 위해서도 필요하다.

### ACKNOWLEDGEMENT

본 연구는 서울시 산학연협력사업의 지원과 지식경제부가 지원하는 국가 반도체 연구개발사업인 시스템 집적반도체기반기술개발사업(시스템 IC 2010)의 지원으로 이루어 졌습니다.

### 참고문헌

- [1] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, no. 6, pp. 1153-1160, 1981.
- [2] T. M. Lehmann, C. Gönner, and K. Spitzer, "Survey: interpolation methods in medical image processing," *IEEE Trans. Med. Imag.*, vol.18, no. 11, pp. 1049-1075, Nov. 1999.
- [3] J. Lee, et. al., "Design of a scan format converter using the bisigmoidal interpolation," *IEEE Trans. Consumer-Electronics*, Vol. 44, No. 3, pp.1115-1121, Aug. 1998.
- [4] E. Aho, J. Vanne, T. D. Hämmäläinen, and K. Kuusilinna, "Block-level parallel processing for scaling evenly divisible images," *IEEE Trans. Circuits and Syst Regular Papers.*, Vol. 52, no. 12, pp.2717-2725