

# 연산량을 감소시킨 선형 Cubic Convolution

## 보간 알고리즘

전영현<sup>0</sup> 윤종호 최명렬  
 한양대학교 전자전기 제어공학  
 {impword<sup>0</sup>, sfw1179, choimy}@asic.hanyang.ac.kr

### Liner Cubic Convolution Interpolation Algorithm with Low Computational Complexity

YoungHyun Jun<sup>0</sup>, JongHo Yun, MyungRyul Choi  
 Dept. of EEIC Hanyang University

#### 요 약

본 논문에서는 Cubic Convolution 보간 알고리즘을 변형하여 연산량을 감소시키고 에지를 강조하는 보간 알고리즘을 제안한다. 제안된 알고리즘은 디지털 영상의 확대 또는 축소에 필요한 연산량을 줄이기 위해 두가지 방법을 사용하였다. 기존의 Cubic Convolution 알고리즘의 고차항의 가중치 연산을 일차원으로 변환하였다. 인접한 픽셀의 차이값을 사용하여 Bilinear 알고리즘을 제한적으로 적용하였다. 제안된 알고리즘의 화질 평가를 위해 원영상의 확대-후-축소와 축소-후-확대를 하여 RMSE를 사용하였고, 연산량을 평가하기 위해 픽셀별 곱셈기와 덧셈기를 기존의 알고리즘과 비교하였다. 시뮬레이션 결과 기존 Cubic Convolution 알고리즘보다 연산량이 감소하였다.

#### 1. 서 론

멀티미디어 산업의 발달과 전자 시스템의 급속한 보급으로 인하여 디스플레이 장치에서 다양한 크기의 디지털 영상 활용은 증가하고 있다. 다양한 크기(QCIF ~ WUXGA)의 디지털 영상을 디스플레이 장치(LCD 모니터, HDTV 등)에 표현하기 위해서는 입력 영상의 해상도를 출력 되는 디스플레이 해상도에 맞게 영상을 조절하는 것이 중요하다. 이러한 보간은 기본적인 신호처리기술 중에 하나이다. 모든 보간의 기본원리는 입력되는 영상의 데이터를 연속적인 데이터로 보고 출력 해상도에 맞게 재표본화(Resampling)하는 것이다.

보간 알고리즘으로는 가까운 값으로 대체하는 Nearest Neighbor[1] 보간 알고리즘이 있으며 가장 간단하게 구현할 수 있지만, 블록 현상(Blocky Appearance)이 발생한다. 일차원 방정식을 사용하는 Nearest Neighbor 보간 알고리즘 보다 성능이 좋은 Bilinear[2] 보간 알고리즘이 있다. 이차원 방정식을 사용하는 Cubic Convolution[3] 보간 알고리즘과 b-spline[2] 보간 알고리즘이 있다.

Cubic Convolution 보간 알고리즘은 많은 연산량을 요구하기 때문에 다양하게 변형된 Cubic Convolution 보간 알고리즘이 나와 있다. 일반적인 보간 알고리즘은 이미지의 화질열화를 발생하며 이러한 화질열화 현상은 예저부분에서 블러링(blurring) 현상으로 나타난다.

본 논문에서는 Cubic Convolution 보간 알고리즘을 변형하여 연산량을 감소시키고 에지를 강조하는 보간 알고리즘을 제안한다. 2장에서는 기존 보간 알고리즘을 소개하고, 3장에서는 제안하는 알고리즘을 설명한다. 4장에서는 시뮬레이션을 통한 알고리즘의 성능을 비교하고 5장에서 결론을 맺는다.

#### 2. 관련 알고리즘

##### 2.1 Bilinear 보간 알고리즘

보간 알고리즘에 사용되는 가장 일반적인 알고리즘이다. 새롭게 생성되는 픽셀값을 인접한 4개의 픽셀값에 가중치를 곱하여 평균으로 생성한다. 가중치는 생성되는 픽셀과 존재하는 픽셀의 거리의 비로 구한다. 3개의 선형 보간이 필요하며, 그림 1은 Bilinear 보간 수행 방법을 나타낸다.

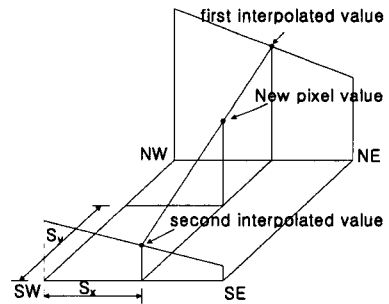


그림 1. Bilinear 보간 수행 방법

##### 2.2 Cubic Convolution 보간 알고리즘

입력되는 영상 데이터를  $g(x)$  라 할 때 많은 보간 함수는 식 (1)로 표현 한다.

$$g'(x) = \sum_i a_i b(x - x_i) \quad \text{식(1)}$$

$g'(x)$ 는 생성될 픽셀값을 나타낸다.  $b$ 는 가중치값이고,  $a_i$ 는 원영상의 픽셀값이다. 식(1)에서  $a_i$ ,  $b$ 는 원영상의 픽셀값( $g(x_i)$ )와

가중치( $f(x_i)$ )로 대체할 수 있다.  $x$ 와  $x_i$ 의 사이의 거리는 식(2)으로 정의 한다.

$$s = x - x_i, \quad 0 \leq s \leq 1, \\ x_i \leq x \leq x_{i+1} \quad \text{식(2)}$$

일차원 Cubic Convolution 보간 함수는 식(3)로 정의한다.

$$f(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1 & 0 \leq |x| < 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & 1 \leq |x| < 2 \\ 0 & 2 \leq |x| \end{cases} \quad \text{식(3)}$$

일차원 Cubic Convolution 보간은 수평보간과 수직보간을 순차적으로 수행하여 이차원 보간을 한다.

그림 2는 일차원 Cubic Convolution 이용하여 이차원 Cubic Convolution 보간 수행 방법을 나타낸다. 먼저 입력되는 영상의 4개 수평라인에 대해 일차원 수평보간을 수행하고, 수평보간된 4개의 값을 수직보간 하여 이차원 보간을 수행한다.

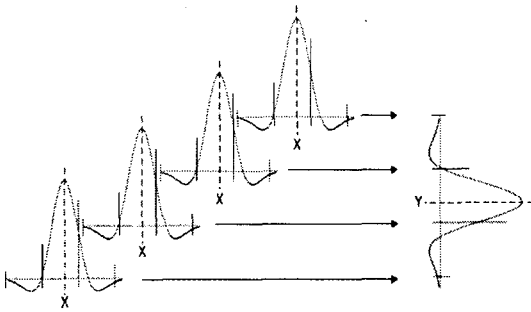


그림 2. 이차원 Cubic Convolution 보간 수행 방법

### 3. 제안된 알고리즘

Cubic Convolution 은 성능에 비해 너무 많은 연산량을 요구한다. 본 논문에서는 영상의 특성을 분석하여 Cubic Convolution 보간에 사용되는 픽셀수를 감소시키는 방법과, 보간 함수를 선형방정식으로 단순화 시켜 연산량을 감소시키는 방법을 제안한다.

Cubic Convolution 보간에는 수직보간과 수평보간에 각각 4개의 가중치를 계산하여야 한다. 각 가중치 마다 3차 방정식의 연산해야 하지만 제안하는 알고리즘에서는 선형 방정식을 사용하여 하나의 곱셈기와 덧셈기를 사용하여 연산량을 감소시킨다.

그림 3과 식(4)은 Cubic Convolution 보간 함수에서  $a=-1$ 일때 선형 Cubic Convolution 함수를 나타낸다.

$$w_x = \begin{cases} -0.375|x| + 1 & 0 \leq |x| < 0.25 \\ -1.25|x| + 1.25 & 0.25 \leq |x| < 1 \\ -0.625|x| + 0.625 & 1 \leq |x| < 1.25 \\ 0.25|x| - 0.5 & 1.25 \leq |x| < 2 \end{cases} \quad \text{식(4)}$$

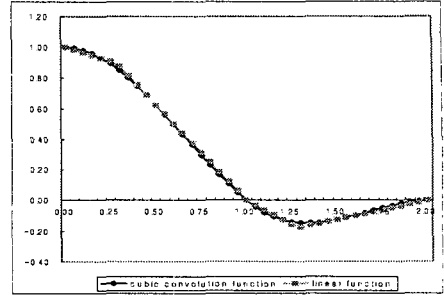


그림 3. 선형 Cubic Convolution 보간 함수

영상의 특성을 분석하면 현재 보간해야 할 위치의 인접 픽셀값의 차이가 크지 않다. 제한된 보간 알고리즘에서는 연산량을 감소시키기 위해 두 가지 알고리즘을 선택적으로 사용한다. 보간해야 할 위치에서 인접의 픽셀값이 차이가 크지 않을 때에는 Bilinear을 사용하여 보간하고, 인접의 픽셀값의 차이가 클때는 선형 Cubic Convolution을 사용하여 보간한다. 식(5)는 인접 픽셀과의 차이값을 구하는 식이며, 그림 4는 인접 픽셀과의 누적 차이값을 나타낸다.

$$Diff(x_k) = abs[a(x_k) - a(x_{k+1})] \\ + abs[a(x_k) - a(x_{k-1})] / 2 \\ + abs[a(x_{k+1}) - a(x_{k+2})] / 2 \quad \text{식(5)}$$

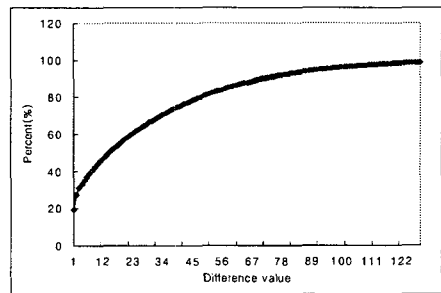


그림 4. 인접픽셀과의 누적 차이값

인접 픽셀값의 차이값을 사용하여 보간 알고리즘을 선택한다. 그림 5는 영상의 확대-후-축소, 축소-후-확대 한 후 RMSE (Root Mean Square Error)를 나타낸다.

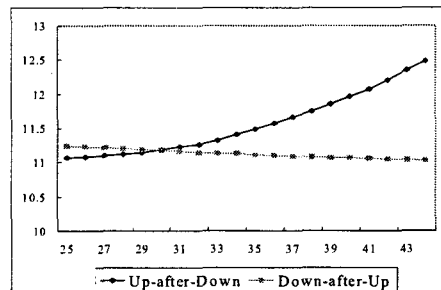


그림 5. 확대-후-축소 와 축소-후-확대 한 후 RMSE

그림 6은 Zoneplate 영상으로 1.6배 확대한 영상을 기존 보간 알고리즘과 비교하였다.

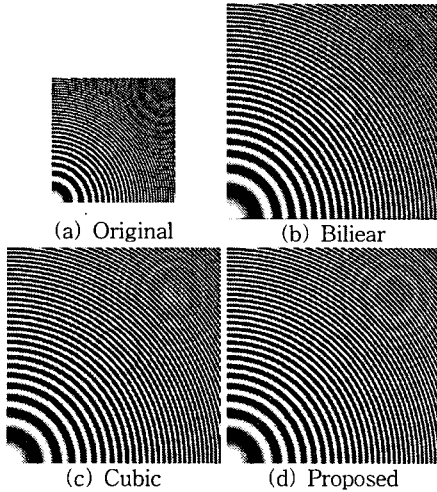


그림 6. Zoneplate 영상의 1.6배 확대한 영상

표 1은 하나의 픽셀에 필요한 연산량을 곱셈과 덧셈으로 나타낸다.

표 1. 필요한 연산량 비교

보간 알고리즘	가중치 연산	참조 픽셀 연산	메모리 연산
Bilinear	곱셈 : 3 덧셈 : 2	곱셈 : 4 덧셈 : 3	읽기 : 4 쓰기 : 1
Cubic	곱셈 : 16 덧셈 : 36	곱셈 : 16 덧셈 : 15	읽기 : 16 쓰기 : 1
Proposed	곱셈 : 7 덧셈 : 13	곱셈 : 8 덧셈 : 7	읽기 : 8 쓰기 : 1

#### 4. 실험 결과

본 논문에서는 화질과 연산량을 기존의 알고리즘과 비교하였다. 화질평가는 RMSE를 사용하였고, 두 가지 방법으로 평가하였다. 확대-후-축소 와 축소-후-확대를 하였다. 확대 비율은 1.6이며 축소비율은 1/1.6이다. 표 2와 표 3은 화질평가를 위해 RMSE를 비교하였다.

표 2. 확대-후-축소 후 RMSE 비교

보간 알고리즘 실험 영상	Bilinear	Proposed	Cubic
F16	2.16	2.05	1.16
Baboon	6.73	4.85	3.57
Lena	2.10	1.95	1.14
Background	9.71	5.61	5.81
Text	12.99	6.45	9.84

표 3. 축소-후-확대 후 RMSE 비교

보간 알고리즘 실험 영상	Bilinear	Proposed	Cubic
F16	5.05	4.12	4.03
Baboon	14.87	14.41	13.92
Lena	4.68	4.06	3.98
Background	22.54	20.14	20.70
Text	29.74	26.11	27.09

그림 7은 기존 알고리즘과 함께 Digital 문자를 확대-후-축소한 영상을 나타낸다.

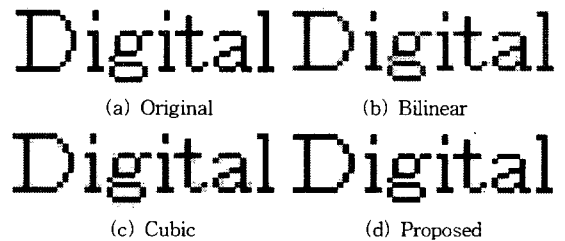


그림 7. Digital 문자를 확대-후-축소한 영상

#### 5. 결론

본 논문에서는 다양한 입력 영상 해상도를 디스플레이 해상도 맞게 변환하는 보간 알고리즘을 제안하였다. 제한된 알고리즘은 기존 Cubic Convolution 보간 함수를 선형 함수로 변환하고, 인접 픽셀값 차이를 이용하여 새로운 픽셀에 필요한 참조 픽셀 수를 감소시켜 연산량을 감소시켰다.

#### 참고문헌

- [1] Randy. Crane, "A Simplified Approach to Image Processing", Prentice Hall, pages 110-143,1997
- [2] R. C. Gonzalez and R. E.Woods, "Digital Image Processing. Reading", MA: Addison-Wesley, 1992.
- [3] Robert G. Keys, "Cubic Convolution Interpolation for Digital Image Processing", IEEE Trans. ASSP, Vol.ASSP-29, pages 1153-1160 (1981)
- [4] C. H. Kim, S. M. Seong, J. A. Lee, L. S. Kim, " Winscale : An Image-Scaling Algorithm Using an Area Pixel Model", IEEE Trans. Circuit and System, Vol.13,No.6, pages 549-553(2003)
- [5] Acciari. G.,Giannini. F., Limiti. E., Rossi. M., "On the Comparison of Bilinear,Cubic Spline,and Fuzzy Interpolation Techniques for Robotic Position Measurements", IEEE Trans. Inst. Meas., Vol.54, No.6, pages 2281-2288(2005)