

# 웨이블릿 기반 압축영상의 화질 향상을 위한 후처리 기법

## Post-processing Technique for Improving Quality of Images in Wavelet-based Image Compression

조 정 현, 유 후, 김 승 종, 정 재 창

한양대학교 전자통신공학과

JungHyun Cho, Hoon Yoo, SeungJong Kim, and JeChang Jeong

Department of Electronic Communications Engineering, Hanyang University

### 요 약

영상 데이터는 방대하므로 한정된 대역폭을 통하여 영상을 전송하기 위해서는 압축이 필요하다. 그러나 압축으로 전송 비트율은 낮아지지만 고압축률에서는 여러가지 화질 열화가 발생한다. 화질 열화에는 이산 여현 변환을 수행하는 압축 부호화 알고리즘에서 나타나는 블록킹 현상 또는 모기잡음이 있고 웨이블릿 기반 압축 영상에서는 에지 부근에 나타나는 링잉 현상이 있다. 이와 같은 화질 열화들은 후처리를 통해 제거되지만, 많은 처리 시간이 소요되지 않도록 해야 한다. 본 논문에서는 웨이블릿 기반 영상 압축에서 링잉 현상을 효율적으로 제거하는 후처리 방법을 제안한다. 제안한 방법에 대한 모의실험 결과 기존의 방법과 비교하여 처리 속도를 빠르게 하면서 유사한 주관적, 객관적 화질을 얻었다.

### 1. 서 론

영상의 전송 및 저장에서 대역폭을 줄이기 위해서 영상 압축을 한다. 이 때 압축으로 비트율은 낮아지지만 고압축률에서는 여러가지 화질 열화가 발생한다. 이런 화질 열화에는 JPEG, H.261, MPEG-1,2, H.263 등 기존의 이산 여현 변환(Discrete Cosine Transform)을 하는 압축 부호화 알고리즘에서 나타나는 모기잡음과 블록킹 현상이 있고 MPEG-4의 정지영상 압축 방법인 웨이블릿 기반 압축에서 발생하는 링잉 현상이 있다. 블록킹 현상은 블록 단위의 양자화 과정에서 생기는 오차 때문에 나타나는 것으로서 압축률이 높아지면 블록 경계에서 불연속성이 마치 타일을 깔 것처럼 규칙적으로 나타나는 화질 열화이다 [1-2]. 링잉 현상(Ringing Artifact)은 고주파수 대역의 계수들이 양자화로 인해 손실됨으로써 에지 부근에 허상의 에지가 보이는 것이다. 링잉 현상은 깁스 현상(Gibbs Effect)이라고도 하며 강한 에지 부근에서 잘 나타난다. 텍스처(Texture) 영역에서는 링잉 현상이 발생하더라도 텍스처 리플(ripple)에 의해 발생한 것인지 아니면 링잉 현상 때문인지 구분이 되지 않는 반면 에지가 있는 평탄(flat)한

영역에서는 링잉 현상이 인간의 시각에 확연히 거슬린다 [3-5].

실시간 처리를 요구하는 응용 분야 등에 있어서는 후처리 할 때 좋은 화질을 갖게 하는 것만큼 시간이 많이 걸리지 않도록 하는 것 또한 중요하다. 따라서 본 논문에서는 효율적으로 후처리하여 후처리 속도를 빠르게 하면서 기존의 방법과 비교하여 비슷한 PSNR과 주관적 화질을 가지는 방법을 제안한다.

본 논문의 II장에서는 링잉 현상을 제거하기 위해 화소 단위로 영상 전체를 후처리하는 기존의 방법을 소개하고, III장에서는 처리 시간을 줄이기 위한 방법으로 에지를 이용한 적응적 후처리 방법을 설명한다. IV장에서는 실험결과를 보이고 마지막으로 V장에서는 결론을 맺는다.

### II. 링잉 현상을 제거하는 방법

Shen이 제안한 후처리 방법[6]은 식(1)에서 식(3)까지로 수행되며 화소 단위로 수행된다. 먼저 화소마다 주변의 값들과 비교하여 대표값  $e$ 를 선택한다.

$$e = \arg \min_{x_i} \sum_{x_j \in N} \rho(x_i - x_j) \quad (1)$$

여기서  $N$ 은 '+'모양을 갖는 9 화소들을 의미하고  $\rho(a)$ 는  $\min\{1/256 \times a^2, 1\}$ 을 사용했다. 그 후 선택된 대표값  $e$ 와 현재의 화소값  $x$ 와의 차  $d$ 를 구한다. 그리고  $d$ 의 절대값이 임계값  $Th1$ 의 2배보다 작은 경우에는 후처리를 수행하고 임계값  $Th1$ 의 2배보다 크면 후처리를 수행하지 않는다.  $y$ 는 현재의 화소값  $x$ 에 후처리 과정으로 교정된 값  $\text{UpDownRamp}(d, Th1)$ 를 합한 값이다.

$$d = e - x \quad (2)$$

$$y = x + \text{UpDownRamp}(d, Th1) \quad (3)$$

이가지  $UpDownRamp(d, Th1)$ 는 식(1)와 같다.

$$UpDownRamp(d, Th1) = Sign(d) * \text{Max}(0, abs(d) - \text{Max}(0, 2*(abs(d) - Th1))) \quad (1)$$

즉 그림 1과 같이  $d$ 의 절대값이  $Th1$ 보다 작은 경우에는 보정치  $UpDownRamp(d, Th1)$ 와의 기울기가 1이고  $Th1$ 과  $2Th1$ 의 사이인 경우에는 기울기가 -1이다.

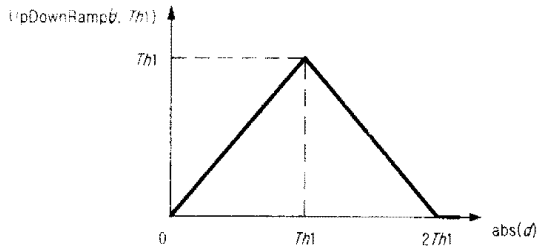


그림 1.  $UpDownRamp(d, Th1)$ 와  $abs(d)$ 의 관계

### III. 적응적 후처리

Shen이 제안한 방법은 화소 단위로 영상 전체를 후처리한다. 따라서 JPEG 2000의 실험 영상 중  $2048 \times 2048$ ,  $2048 \times 2560$  크기를 갖는 Aerial2, Bike 등과 같은 경우에는 후처리에 많은 시간이 소요된다. 그림 2와 같이 Shen의 방법에서 후처리 후 화소값이 바뀐 화소들을 조사해보면 주로 고주파수 성분이 많은 에지 부분과 텍스처(Texture) 부분이라는 것이 확인된다. 따라서 본 논문에서는 후처리 전에 후처리 과정을 거쳐도 변화되지 않는 부분을 미리 제거함으로써 처리 시간의 단축을 가져오는 방법을 제안한다.



그림 2. Shen 방법에서 후처리되는 화소의 Bike 부분 영상

그림 3,4와 같이 에지 검출 후 후처리하는 블록을 구해보면 그림 2의 후처리하는 화소들이 존재하는 부분과 비

슷한 것을 확인할 수 있다. 실질적으로 영상 현상이 나타나는 부분 중 일간 시각이 화질 열화로 느껴지는 부분이 에지 부분인 것을 고려해 볼 때 영상 중 에지 부분만 후처리하는 것은 타당하다.

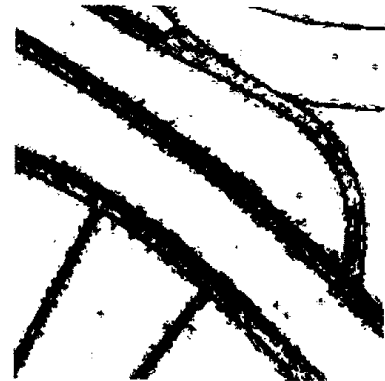


그림 3. 에지로 검출된 화소의 Bike 부분 영상

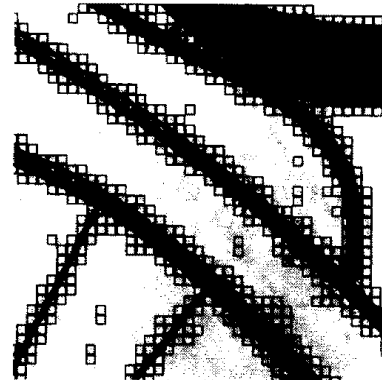


그림 4. 후처리되는 블록의 Bike 부분 영상

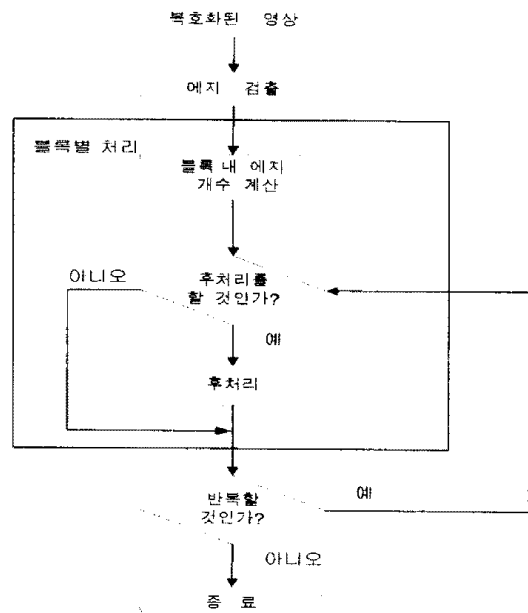


그림 5. 제안하는 알고리즘 구성도

그림 5는 제안한 적응적 후처리 과정의 알고리즘이다. 복호화 과정을 거친 영상을 이용하여 에지를 검출하고 검출된 에지 정보를 이용하여 블록 단위로 후처리할 영역을 선정하여 전체적인 영상 모두에 대해 후처리를 하지 않고 링잉 현상이 발생하는 부분에만 후처리를 하도록 하였다.

에지를 검출하는 방법으로는 여러 가지가 연구되고 있으나 비교적 에지를 잘 찾으면서 에지 검출 시간이 적은 소벨 연산자(Sobel Operator) 방법을 이용하였으며 그 때의 임계값은 100으로 하였다. 그리고 후처리할 블록을 찾을 때 블록의 크기는  $8 \times 8$  로 하였으며 영상이 큰 경우에는  $16 \times 16$  또는  $32 \times 32$ 로도 가능하다. 후처리할 블록을 결정하는 기준은  $8 \times 8$  블록인 경우에 블록 내의 에지로 찾은 화소의 개수가 1개 이상인 경우에는 후처리하도록 하였다.  $8 \times 8$  블록 중 에지로 찾은 화소가 없는 경우가 전체 블록 개수 중 반 정도를 차지하고 있으므로 에지로 찾은 화소가 없는 경우만 제외시키더라도 후처리하는 시간을 줄일 수 있다.

#### IV. 실험 결과

제안한 방법을 이용하여 다섯 개의 실험 영상을 가지고 실험을 하였다. 실험에 사용된 영상은  $512 \times 512$  크기의 Target 영상,  $2048 \times 2048$  크기의 Aerial2 영상, 그리고  $2048 \times 2560$  크기의 Bike, Cafe, Woman 영상이다. 실험에서 이용된 *Tk1* 값은 0.25bpp, 0.125bpp, 0.0625bpp일 때 각각 8, 10, 12이다.

표 1-6은 0.25bpp, 0.125bpp, 0.0625bpp일 때 각 영상의 PSNR 결과이고 그림 6-9은 0.125bpp일 때의 후처리한 결과의 부분 영상이다. 전체적으로 제안한 방법은 기존의 방법과 비슷한 PSNR을 가지면서 계산량이 반으로 줄어든다. 그러나 Aerial2, Woman에서는 PSNR이 떨어진다. 왜냐하면 Aerial2가 자연 영상을 촬영한 항공사진으로써 강한 에지보다는 텍스처 부분이 많아 효과를 보지 못하는 것이고 Woman에서도 텍스처 부분인 사람의 옷을 후처리함으로써 PSNR이 떨어짐을 알 수 있다. 그림 3-6에서 에지 부근에 보이는 링잉 현상이 제거되어 주관적 화질도 좋아졌음을 확인할 수 있다. 표 7에는 후처리하는 블록의 개수를 나타내었다. 전체 블록 수 중 후처리 하는 블록 수가 반 정도로 줄어들어 처리 시간을 줄였다.

표 1. 0.25bpp일 때 기존의 방법

	PSNR	후처리 후의 PSNR		
		I=1	2	3
Aerial2	28.700630	28.470072	28.251991	28.077536
Bike	28.298422	28.498827	28.535188	28.511395
Cafe	23.216454	23.303621	23.319909	23.308810
Target	22.175488	22.229667	22.255797	22.273208
Woman	29.778166	29.653083	29.438778	29.232541

표 2. 0.25bpp일 때 제안한 방법

	PSNR	후처리 후의 PSNR		
		I=1	2	3
Aerial2	28.700630	28.463877	28.250191	28.081804
Bike	28.298422	28.460565	28.481243	28.451358
Cafe	23.216454	23.293659	23.304692	23.290719
Target	22.175488	22.219623	22.239162	22.251889
Woman	29.778166	29.639450	29.424528	29.220695

표 3. 0.125bpp일 때 기존의 방법

	PSNR	후처리 후의 PSNR		
		I=1	2	3
Aerial2	26.638126	26.534769	26.423145	26.325342
Bike	25.687823	25.851120	25.921117	25.946397
Cafe	20.752382	20.798795	20.810082	20.805327
Target	19.651409	19.677675	19.689434	19.695593
Woman	27.252525	27.138104	27.020892	26.898484

표 4. 0.125bpp일 때 제안한 방법

	PSNR	후처리 후의 PSNR		
		I=1	2	3
Aerial2	26.638126	26.521048	26.407440	26.311215
Bike	25.687823	25.819862	25.870486	25.884063
Cafe	20.752382	20.791571	20.798133	20.790159
Target	19.651409	19.672253	19.680159	19.683327
Woman	27.252525	27.125796	27.005536	26.883582

표 5. 0.0625bpp일 때 기존의 방법

	PSNR	후처리 후의 PSNR		
		I=1	2	3
Aerial2	24.714996	24.688425	24.647717	24.610458
Bike	23.053943	23.149309	23.199716	23.227507
Cafe	19.104579	19.128452	19.134862	19.131306
Target	17.356186	17.361452	17.377476	17.383716
Woman	25.459373	25.359868	25.285389	25.205061

표 6. 0.0625bpp일 때 제안한 방법

	PSNR	후처리 후의 PSNR		
		I=1	2	3
Aerial2	24.714996	24.675292	24.628869	24.588378
Bike	23.053943	23.128843	23.164581	23.181672
Cafe	19.104579	19.121202	19.122430	19.115170
Target	17.356186	17.356975	17.369794	17.373493
Woman	25.459373	25.350793	25.272271	25.190572

표 7. 제안한 방법의 후처리 블록(8×8) 개수

	총 블록	0.25 bpp	0.125 bpp	0.0625 bpp
Aerial2	65536	49931	38703	31042
Bike	81920	36763	32542	28613
Cafe	81920	57622	54815	47835
Target	4096	2010	1918	1688
Woman	81920	35947	28788	22529



그림 6. 후처리 전의 Bike의 부분 영상



그림 7. 후처리 후의 Bike의 부분 영상



그림 8. 후처리 전의 Cafe의 부분 영상



그림 9. 후처리 후의 Cafe의 부분 영상

## V. 결론

본 논문에서는 웨이블릿 기반 영상 압축에서 발생하는 링잉 현상을 제거를 위한 후처리 기법을 적용적으로 구현하기 위하여 에지 정보를 이용하였다. 제안한 방법은 기존의 방법과 비교하여 유사한 주관적, 객관적 화질을 얻으면서 계산량을 줄일 수 있었다.

향후 에지 부근 이외의 부분에서의 화질 향상을 위한 후처리 연구 및 링잉 현상 외의 웨이블릿 기반 부호화에 서 발생하는 화질 열화들을 제거할 수 있는 방법의 연구가 필요하다.

## VI. 참고문헌

- [1] B. Jeon, J. Jeong, and J. Jo, "Blocking artifacts reduction in image coding based on minimum block boundary discontinuity", *Visual Commun. Image Processing '95*, pp. 198-209, May 1995.
- [2] J. Jeong and B. Jeon, "Use of a class of two-dimensional functions for blocking artifacts reduction in image coding", *IEEE International Conference on Image Processing*, pp. 478-481, Oct. 1995.
- [3] G. Strang and T. Nguyen, "Wavelets and Filter Banks", Wellesley-Cambridge Press, 1996.
- [4] Ramon Llados-Bernaus, and Robert L. Stevenson, "Edge Assisted Upper Bands Coding Techniques", *SPIE Vol. 3309*, pp. 2-13.
- [5] M. Vetterli and J. Kovačević, "Wavelets and Subband Coding", Prentice Hall PTR, 1995.
- [6] Meiyin Shen, "Postprocessing for removing wavelet coding artifacts," ISO/IEC JTC1/SC29/WG1N935, June 1998.