2000年度 第13回 信號處理合同學術大會 論文集 第13卷 1號

메모리를 반으로 줄이는 간이형 비디오 디코딩 알고리즘

이원백*, 이동호*

*한양대학교 전자컴퓨터공학부

A Video Decoding Algorithm Which Reduces the Frame Memory to Half

Won-Baek Lee*, Dong-Ho Lee*

*School of Electrical and Computer Engineering of Hanyang Univ

wblee@image.hanyang.ac.kr, dhlee@image.hanyang.ac.kr

요 약

본 논문은 완전한 규격의 비디오 디코더를 구현하는 것이 아니라 하드웨어의 메모리를 절반으로 줄인 비디오 디코더에 관한 것이다. 우선 프레임 메모리를 수평 방향으로 1/2 만큼 축소시키는 방법을 제시하고, 다음으로 이렇게 축소되어 저장된 프레임 메모리를 움직임 보상을 하기 위해 다시 Interpolation 하는 방법을 제시한다. 이 때 여러 방법의 모의 실험을 통해 추출된 영상의 특징들을 이용하여 메모리를 줄였을 때 나타나는 화질의 열화와 에러의 누적을 최소화하는 적응적인 알고리즘을 제시하고, 컴퓨터 모의 실험을 통해 기존에 사용하던 방법과 비교하여 제안된 알고리즘의 성능을 검증하고 결론을 맺는다.

I. 서 론

최근에 차세대 가전제품으로 DTV 에 대한 연구가 활발히 이루어지고 있다. 이러한 DTV 의 영상 압축 규격은 MPEG-2(ISO 13818)의 MP@HL 을 따르고 있는데 이에 따라 DTV 비디오 디코더를 구현할 경우 적어도 4-5 이상의 Frame Memory 가 필요하다.

본 논문은 완전한 규격의 비디오 디코더를 구현하는 것이 아니라 프레임 메모리를 절반으로 줄인 간이형 비디오 디코더에 관한 것으로 메모리를 줄이게 되면 화질의 열화가 생기게 되는데 이렇게 화질이 떨어진 영상을 사용하여 움직임보상을 하게 되면 디코딩이 진행 될수록 화질은 급격하게 나빠지게 된다. 그로 인해 발생하는 문제들을 최소화 시키는 비디오 디코더의 구현 방법 및 구현시에 문제가 될 수 있는 부분을 보완하는 알고리즘을 제안한다. 이 방법은 Interpolation 필터의 설계가 가장 핵심적인 부분으로 기존의 알고리즘이 공간 필터의 한계, 즉 빠른 움직임이나 급격한 변화를 많이 포함한 영상의 경우 출력의 품질을 보장하기 어렵기 때문에 이 부분을 잘 보완할 수 있는 Interpolation 필터의 설계는 매우 중요하다. 본 논문에서는 움직임등의 급격한 변화를 찾아내기 위해 Edge 를 기반으로 한다. 또 Edge 가 아닌 부분에서는 주변 픽셀들의 변화와 상관성 등을 사용하여 Interpolation 하는 효율적이고 향상된 성능의 알고리즘을 제안한다.

II. 본 론

1. 제안 알고리즘

본 알고리즘은 IDCT 의 블록 크기를 8X8 로 고정하여 모든 Coefficient 를 사용하여 디코딩을 하고 프레임 메모리에 넣기 전에 수평 방향으로 1/2 만큼 줄여 저장하므로 메모리의 크기를 1/2 로 줄일 수 있는 것이다. 이렇게 저장된 메모리를 Interpolation 하여 움직임 보상을 할 때 사용하는 알고리즘이다. Interpolation 된 영상이 움직임 보상을 할 때 사용되므로 Interpolation 의 결과가 디코딩에 계속 영향을 미치게 된다. 따라서 Interpolation Filter 의 성능에 따라 출력 영상의 품질이 결정되는 것은 물론이고 Error 의 Propagation 에도 많은 영향을 미치게 된다.

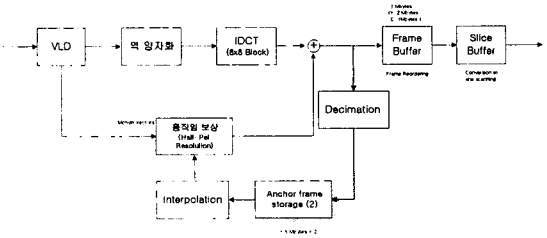
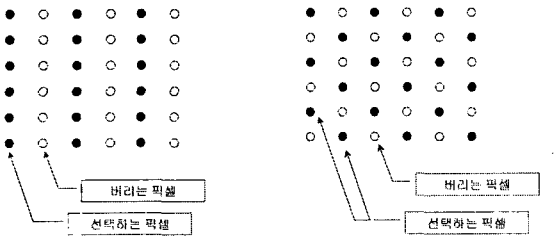


그림 1. 제안된 디코딩 알고리즘

2.2 Decimation



(a) Rectangular 방법 (b) Interlaced 방법

그림 2. Decimation 방법

프레임 메모리에 저장된 영상은 그대로 사용될 것이 아니고 Interpolation 을 위해 사용되므로 원래의 영상이 왜곡되지 않고 정확한 픽셀 값을 확보할 수 있어야 한다. 그리고 Interpolation 할 때에 수평 뿐만 아니라 수직 방향으로도 참고할 픽셀을 가지는 Decimation 방법이 필요하였다. 이와 같은 조건들을 만족시킬 수 있는 방법으로 홀수 라인에서는 홀수열의 픽셀을 취하고 짝수열의 픽셀은 버리며 짝수 라인에서는 짝수열의 픽셀을 취하고 홀수열의 픽셀은 버리는 Interlaced Subsampling 방법을 사용하였다..

2.3 Interpolation

Edge 영역의 Interpolation

일반적으로 영상의 Edge 를 찾기 위해서는 메모리에 저장된 영상에 x_operator 와 y_operator 를 적용하여 둘 중 어느 하나가 Threshold 값보다 클 경우 Edge 로 간주하고 Edge 로 결정된 부분은 Edge 의 방향에 따라 Interpolation 하게 된다[4][5].

그런데 본 논문에서는 Edge 를 찾아내기 위해 메모리에 저장된 영상을 사용하는 것이 아니라 Zero Padding 영상, 즉 Interpolation 해야 할 부분에 " 0" 값을 넣은 영상을 사용하였다. Zero Padding 영상을 사용할 경우 3x3 Sobel Operator 를 사용하면 실제 계산에 사용되는 픽셀은 좌우 두 픽셀 또는 상하 두 픽셀 밖에 사용되지 않는다. 따라서 Operator 의 확장이 필요했고 그 결과 5x5 Sobel Operator 를 사용하게 되었다. 이 operator 를 사용해서 모의 실험을 해보았더니 Edge 도 더 정확하게 찾아 내었고 메모리에 저장된 픽셀, 즉 Interpolation 할 주변의 픽셀뿐만 아니라 Interpolation 할 픽셀도 Edge 인지 아닌지도 알 수 있게 되었다. 또 보다 정확한 Interpolation 을 하기 위해 수평 방향과 수직 방향의 Edge Map 을 따로 만들어 Interpolation 할 픽셀이 Edge 이면 수평 방향 Edge 인지 수직 방향 Edge 인지를 조사하여 그 방향에 따라 Interpolation 하도록 하였다.

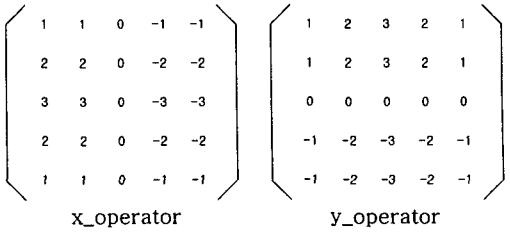


그림 3. 5x5 Sobel Operator

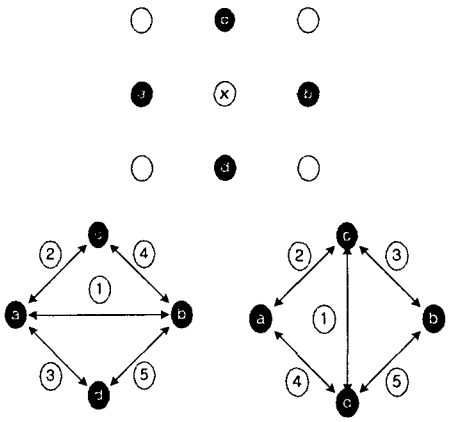
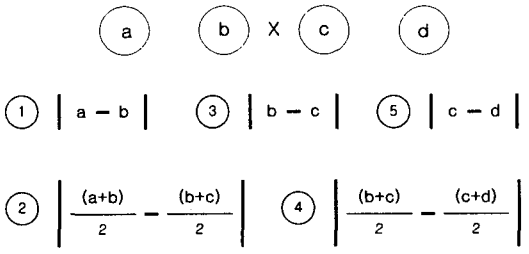


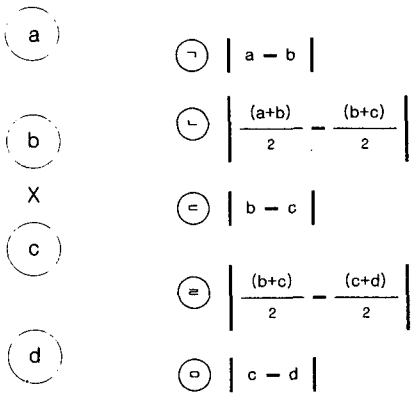
그림 4. Edge 부분의 Interpolation

Correlation Detection 알고리즘

아래 그림에서 X 는 Interpolation 할 픽셀이며 주변 픽셀들의 값을 사용하여 주변 픽셀들간의 Correlation 을 구한다. 가로 방향의 픽셀 값 ①,②,③,④,⑤ 중에서 4 개 이상이 Threshold 값 보다 작으면 가로 방향으로 Correlation 이 있다고 보고, 세로 방향의 값 (ㄱ),(ㄴ),(ㄷ),(ㄹ),(ㅁ)중에서 4 개 이상이 Threshold 값 보다 작으면 세로 방향으로 Correlation 이 있다고 본다. 세로 방향 연관성에서 ⓐ와 ⓓ는 존재하지 않는 값으로 좌우 픽셀 값의 평균을 취하여 사용한다.



(a) 가로 방향 연관성



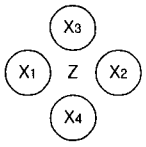
(b) 세로 방향 연관성

그림 5. Correlation Detection Algorithm

Median Filter Method

Median Filter 란 주어진 N 개의 값들 중 중간의 값을 취하는 필터로서, 여기서는 상하좌우 주변 4 개의 픽셀과 가로방향으로 4 개 픽셀을 FIR 필터를 씌운 값, 세로방향으로 4 개 픽셀에 FIR 필터를 씌운 값, 그리고 주변 8 개 픽셀에 FIR 필터를 씌운 값을 Median 필터를 사용하는 방법이다[3]. 이 방법은 FIR Filter 를 사용하므로 주변 픽셀끼리 평균을 하므로 영상 전체를 부드럽게 해주는 효과가 있다. 하지만 주변의 픽셀 값들의 차이가 클 경우에는 실제의 픽셀 값과는 상당한 오차가 있을 수 있다는 문제점이 있다.

본 논문에서는 앞에서 말했던 픽셀 값의 오차를 줄이기 위해 주변 픽셀간의 연관성을 이용하여 필터에 넣는 값들이 적응적으로 변하게 된다. 그 방법을 그림 6 에 나타내었다.



Z = Med { X1, X2, XHAVE, XHFIR, XFIR }
(a) 수평 방향성이 있을 경우의 픽셀 선택

Z = Med { X3, X4, XVAVE, XVFIR, XFIR }
(b) 수직 방향성이 있을 경우의 픽셀 선택

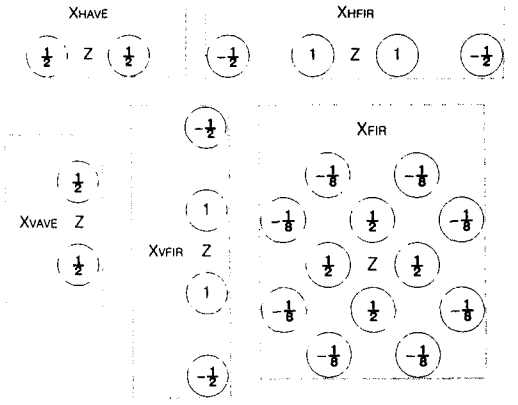


그림 6. Proposed Median Filter Structure

전체 Interpolation 알고리즘

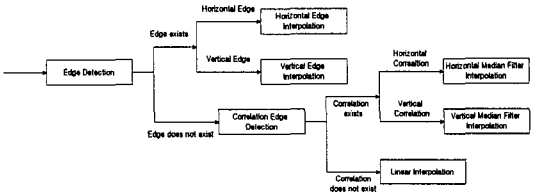


그림 7. Proposed Interpolation Algorithm

지금까지 몇 가지의 Interpolation 알고리즘을 제안하였는데 이것들을 영상의 특성에 따라 적응적으로 적용시켜 Interpolation 하게 하였다. 우선 Zero Padding 된 영상에 5X5 Sobel Operator 를 실행시켜 Edge 를 찾아내고 Interpolation 하는 픽셀이 Edge 인지 아닌지를 확인한다. Edge 인 경우 그것이 가로 방향의 Edge 인지 세로 방향의 Edge 인지에 따라 방향에 맞게 Edge Interpolation 을 하고 Edge 가 아닌 경우에는 주변 픽셀들간의 연관성을 찾아 연관성이 있으면 그것이 가로 방향의 연관성인지 세로 방향의 연관성인지를 선택한다 그리고 주변 픽셀과의 연관성에 따라 적응적으로 선택된 픽셀 값을 Median Filter 를 거쳐서 선택한다.

III. 모의 실험 결과

메모리를 반으로 줄인 간이형 디코더의 모의 실험에 사용된 영상은 720X480 의 해상도를 가진 ML 급의 Ballet 영상과 1920X1088 해상도를 가진 HL 급 Boxing 영상이다. 모두 비월 주사식 영상으로 Frame Coding 이 된 것을 사용하였다. 제안된 알고리즘의 성능 평가를 위해 기존의 8x4 IDCT 를 사용하는 디코딩 알고리즘과 함께 실험하였다[2].

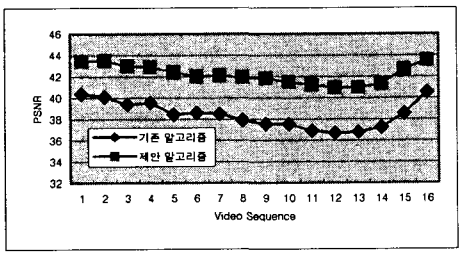
Video sequence	1	2	3	4	5	6	7	8	9
Picture Type	I	B	B	P	B	B	P	B	B
기존 알고리즘	40.38	40.12	39.40	39.56	38.46	38.60	38.50	37.93	37.57
제안 알고리즘	43.44	43.47	42.99	42.88	42.39	42.00	42.09	41.94	41.83
Video sequence	10	11	12	13	14	15	16	Average	
Picture Type	P	B	B	P	B	B	I		
기존 알고리즘	37.56	36.89	36.70	36.61	37.24	36.57	40.56	38.43	
제안 알고리즘	41.46	41.20	40.93	40.97	41.34	42.68	43.52	42.20	

(a) Ballet Image

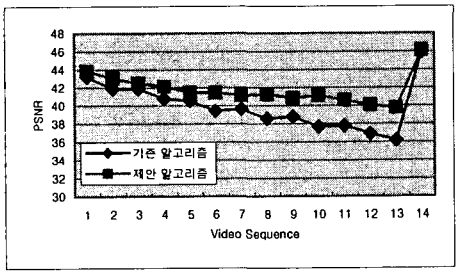
Video sequence	1	2	3	4	5	6	7	8	9
Picture Type	I	B	P	B	P	B	P	B	P
기존 알고리즘	43.23	41.89	41.86	40.77	40.54	39.42	39.64	38.55	38.76
제안 알고리즘	43.85	43.02	42.50	42.12	41.46	41.43	41.15	41.26	40.86
Video sequence	10	11	12	13	14	Average			
Picture Type	B	P	P	P	I				
기존 알고리즘	37.65	37.78	36.91	36.18	45.99	39.94			
제안 알고리즘	41.14	40.50	40.06	39.78	46.14	41.82			

(b) Boxing Image

표 I. 각 영상의 PSNR



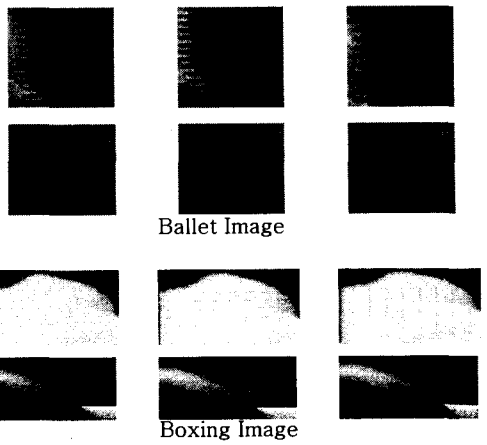
(a) Ballet Image



(b) Boxing Image

그림 8. 각 영상의 PSNR 의 비교

위의 결과를 보면 평균 PSNR 값이 Ballet 영상에서 3.8dB, Boxing 영상에서 1.9dB 정도 상승하였고 Error 의 Propagation 폭도 각각 1.3dB 와 3.4dB 가 줄어드는 등 제안 알고리즘의 객관적인 성능의 향상을 알 수 있었다.. 또 주관적인 성능의 평가를 병행하기 위해서 두 가지 알고리즘의 출력 영상을 눈으로 직접 비교해 보았는데 역시 본 논문에서 제안한 알고리즘에서 더 낳은 영상을 볼 수 있었다.



Ballet Image

Boxing Image

(a)원영상 (b)제안 알고리즘 (c)기존 알고리즘

그림 9. 출력 영상의 비교

IV. 결 론

본 논문에서는 프레임 메모리를 반으로 줄이는 효율적인 비디오 디코딩 알고리즘을 제안하였다. 메모리를 줄이고 다시 복원하는 과정에서 효과적인 Interpolation 을 하기 위하여 영상의 특징에 따라 적응적으로 Interpolation 되도록 하여 원 영상에 최대한 근접한 영상을 만들도록 하였고 컴퓨터 모의 실험을 통해 제안하는 알고리즘이 기존의 방법에 비해 PSNR 의 상승 뿐만이 아니라 Error 의 누적 폭을 상당히 줄이는 등 상당한 성능의 개선을 보여주었다. 또 주관적인 평가를 위해 직접 출력 영상을 비교한 결과도 기존의 방법이 갖는 Blocking Artifact 나 영상에 오류가 생기는 문제점들을 충분히 해결하고 있다.

V. 참고 문헌

[1] ISO 13818-2:1994 *Information technology – Coding of Moving Pictures and Associated Audio – Part 2 : Video*

[2] Dongho Lee, "HDTV Video Decoder Which Can Be Implemented With Low Complexity", *International Conference on Consumer Electronics*, June 1994

[3] Bing Zeng and Anastaslos N. Venetsanopoulos, "Image interpolation based on median-type filters," *Optical Engineering*, Vol. 37, No. 9, September 1998.

[4] Nicos Herodotou and Anastaslos N. Venetsanopoulos, "Color image interpolation for high resolution acquisition and display device," *IEEE Transactions on Consumer Electronics*, Vol. 41, No. 4, November 1995.

[5] J. Allebach and P. W. Wong, "Edge-detected interpolation," *IEEE Int. Conf. Image Processing*, September 1996.