

De-interlacing을 위한 효과적인 시/공간 보간 알고리즘

이성규*, 이동호*

(* : 한양대학교 전자컴퓨터 공학부)

An Efficient Spatial and Temporal Interpolation for Adaptive De-interlacing

Sungku Lee*, Dongho Lee*

(* : School of Electrical and Computer Engineering of Hanyang Univ.)

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

요 약

본 논문에서는 효과적인 De-interlacing을 위한 Edge based Median Filter와 3-Step AMPD(Adaptive Minimum Pixel Difference Filter)를 제안한다. Motion Adaptive De-interlacing 방법에서 중요한 요소인 Motion Missing에 의한 에러를 방지하기 위해 입력 영상을 4 가지 유형으로 구분하여 각 영상에 따라 다른 임계 값을 적용하여 정확한 화소 값을 보간 하는 AMPD(Adaptive Minimum Pixel Difference) Filter를 사용하며 Moving Diagonal Edge의 효과적인 보간을 위해서 방향 필터를 사용하여 Edge Map을 추출한 뒤 Edge에 따라 가변적인 후보 화소를 선택하는 Edge based Median Filter를 사용하여 성능을 향상시켰다. 또한 입력되는 영상을 움직임 영역, 정지 영역, 경계 영역으로 나누어 적응적으로 보간 하여 연산 효율을 높였다. 제안된 방법은 다양한 영상에 대한 모의실험을 통해 기존의 방법에 비해 뛰어난 성능 개선을 보였다.

I. 서 론

현재 HDTV 시스템의 개발에서 Interlace to Progressive 변환(IPC) 시스템은 HDTV에서 채택된 다양한 표준 포맷을 지원하기 위해 필수 불가결한 요소가 되었다. 지금까지 다양한 방법들이 제시되었으며 이는 크게 MC(Motion Compensation) 방법과 No-MC(No Motion Compensation) 방법으로 나눌 수 있으며[1], No-MC 방법에는 다양한 공간 필터를 사용한 접근 방법과 움직임에 따라 적응적으로 변하는 (Motion-Adaptive) 방법 등이 있다. 널리 사용하는 방법으로 Edge를 추출하여 이를 이용하여 라인간 평균값을 구하는 ELA(Edge based Line Average)방법이 간단한 계산과 구현의 용이성 때문에 많이 사용되고 있지

만 작은 화소 값의 변화에 민감하고 잘못된 Edge 정보를 이용하는 등의 단점이 있으며[2][3], 움직임 정보를 이용한 MC방법은 잘못된 움직임 추정 시 에러가 전달되며 구조가 복잡하다[4][5]. Motion Adaptive 방법에 있어서 중요한 부분은 먼저 정확한 움직임 정보의 추출과 Edge Map의 추출, 마지막으로 효율적인 시-공간 보간 필터의 설계이다. 일반적으로 움직임을 얻기 위해서 3 개의 필드에서 화소 값의 차이를 비교하여 화소 값의 차이가 임계치보다 큰 경우에 움직임이 있는 화소 값으로 취하게 되는데 이 경우 정확한 움직임 정보를 얻는데 한계가 있으며 이렇게 발생된 Motion Missing부분에 있어서는 급격한 화질의 열화를 가져온다. 필터의 성능을 좌우하는 요소로서 Motion Missing과 False Motion이 있는데 Motion Missing이란 움직임이 있는 영역을 감지하지 못해 생기는 에러이고 False Motion이란 움직임이 없는 영역을 움직임이 있는 영역으로 간주하여 발생하는 에러이다. 공간 필터의 사용은 먼저 Edge Map을 추출하여 이에 따라 적절한 화소 값을 취하게 되는데 이 경우 정확한 Edge Map의 추출이 성능을 좌우하는 중요한 인자이다. Edge를 추출하기 위한 다양한 방법들이 제시되었는데 대부분의 방법에 있어서 대각선 Edge에 취약한 단점이 있고 구해진 Edge의 방향을 완전히 신뢰할 수 없다. Temporal Filter만을 사용하는 경우에는 빠르게 움직이는 영상이나 급격한 화면의 변화에 대해서 화질의 열화를 가져온다. 그러므로 이러한 요소들의 성능을 전반적으로 개선하여 효율적인 알고리즘을 제안한다.

II. 본 론

1. 제안 알고리즘의 구성

그림 1은 제안 알고리즘의 블록 다이어그램이다. 제안 알고리즘을 블록별로 설명하면 Motion Detection 블록

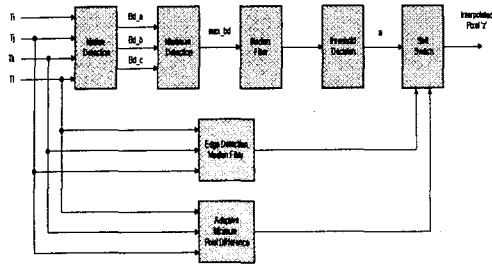


그림 1. 제안 알고리즘 블록도

에서는 3개의 Reference Field로부터 입력되는 화소 값을 이용하여 움직임 영역을 판별하기 위해 각 화소 값의 차이를 구하고 Maximum Detection 블록에서는 3개의 후보 화소 차를 이용하여 이 중 최대 화소 차를 움직임 값으로 정하여 움직임 영역을 판별하게 된다. 잡음 제거를 위해서 Median Filter를 통과하여 구해지는 움직임 값을 이용하여 Threshold Decision 블록에서는 먼저 입력 영상의 움직임 정도를 파악하여 크게 4가지의 영상으로 분류하며 움직임 정도에 따라 AMPD(Adaptive Minimum Pixel Difference) Filter에서 사용될 임계 값과 Softswitch 블록에서 사용될 임계 값을 결정한다. Softswitch 블록에서는 이 값을 이용하여 Spatial Filter를 통과한 값을 사용할 지, Temporal Filter를 통과한 화소 값을 사용할 지, 혼용하여 사용할 지를 결정한다. Moving Diagonal Edge 부분의 효율적인 보간을 위해서 Edge based Median Filter가 사용되며 Motion Missing을 제거하기 위해서 주위 화소 값을 3 단계로 고려하여 화소 값을 결정하는 AMPD Filter를 사용한다. Brightness Difference를 사용하므로 False Motion은 발생하지 않는다

2. 움직임 감지 및 임계값의 결정

그림 2는 'z'위치에 있는 화소의 움직임 값을 구하기 위해 사용되는 그림이다. 이렇게 구한 값 중에서 최대값을 움직임 값으로 정한다. 밝기 차를 이용하여 움직임 값을 구할 때 특정 부분에서의 미세한 움직임은 감지되는 않는 단점이 있다. 이러한 특정 부분에서 발생하는 False Motion의 경우 AMPD(Adaptive Minimum Pixel Difference) 필터를 통해 보상하므로 화질의 열화는 발생하지 않는다. 이러한 움직임 정보는 노이즈

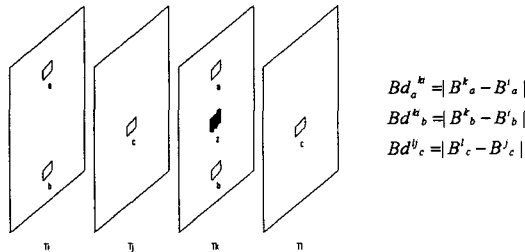


그림 2. 3-D Motion Detection

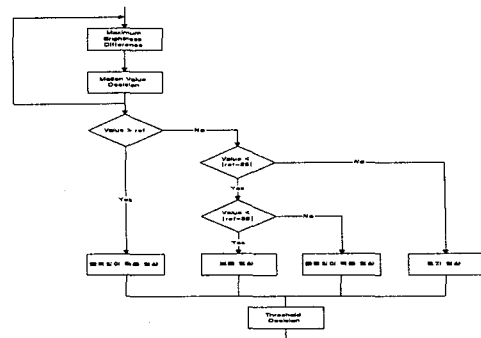


그림 3. 입력 영상의 분류

제거를 위해 Median 필터를 통과시킨다. 구해진 움직임 정보를 이용하여 입력되는 영상을 크게 움직임이 많은 영상, 보통 영상, 움직임이 적은 영상, 정지 영상의 4 가지 유형으로 분류하고 이렇게 구해진 영상을 다시 부분별로 움직임이 있는 영역, 배경 영역, 경계 영역으로 세분한다. 움직임이 많은 영상의 경우 보간될 화소 값에 대한 정보가 공간상에 더 많이 존재하므로 공간 필터에 중점을 두어 보간하고 정지 영상의 경우 시간 축에 더 많은 화소 정보가 존재하므로 Temporal 필터에 더 비중을 두어 보간 한다. 그러므로 입력되는 영상의 종류에 따라 임계 값을 가변적으로 변화시킴으로써 정확한 보간이 이루어지게 하며 불필요한 부분의 연산을 줄임으로서 연산의 효율성을 높인다. 그림 3은 입력되는 영상을 4 가지 타입으로 분류하고 임계 값을 결정하는 흐름도이다.

3.Spatial Interpolation(Edge based Median Filter)알고리즘의 성능을 좌우하는 중요한 문제 중의 하나는 Moving Diagonal Edge의 처리이다. 일반적인 화소 차이 값이나 Median Filter를 이용할 경우 가는 Edge가 존재할 때 이 화소 값의 보간에서 잘못된 화소 정보를 이용하는 단점이 있고 기존의 Edge based De-interlacing Method의 경우 45, 0, -45 도 방향의 Edge의 처리는 가능하지만 더 완만하게 변화하는 Edge의 경우 화질의 열화를 가져온다. 여기서는 Edge based Median Filter를 제안한다. 이는 두 단계로 나누어지는데 1차적으로 3*3 Sobel 연산자를 사용하여

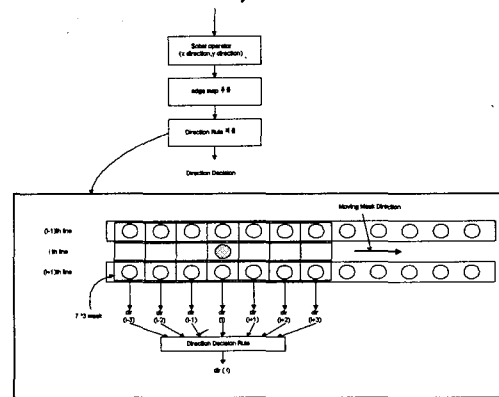


그림 4. Edge map 추출을 위한 흐름도

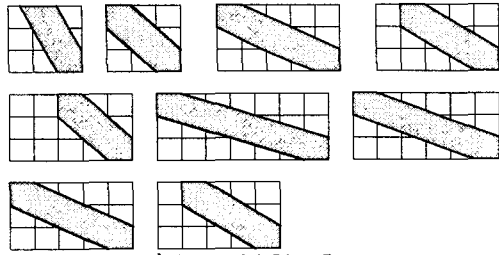


그림 5. Spatial Line Pattern

Edge Map을 추출하고 일반적인 Window보다 큰 7*3 Window를 선택하여 완전한 Moving Diagonal Edge에 대한 성능을 향상시켰다. 방향성을 찾기 위해서 수평 방향과 수직 방향으로 각각 연산자를 적용하여 수평 방향 Edge Map과 수직 방향 Edge Map을 추출한다. 이는 일반적인 3*3 Window가 45, 0, -45도 방향의 Edge Map을 추출할 수 있는데 비해 7*3 Window를 적용함으로써 45, 30, 15, 0, -15, -30, -45 도 방향의 완전한 Edge에 대해서도 성능을 발휘한다. 그림 4는 Edge의 방향을 최종적으로 추출하기 위한 흐름도이고 그림 5는 가능한 공간 라인 패턴을 나타낸 그림이다 그림 5는 한 방향에 대해서 가능한 패턴이며 같은 방법으로 반대 방향과 180도 반대 방향에 대해서도 적용한다. 점선은 보간된 화소를 의미한다.

4. Temporal Interpolation(3 - Step Adaptive Minimum Pixel Difference Filter)

Motion Missing에 의해 발생하는 에러를 제거하기 위해 3 단계 탐색 과정을 거친다. 3 단계 탐색 과정의 전제 조건은 다음과 같다.

- 1) 움직임이 감지되는 얇은 부분, 즉 Motion Missing이 발생하는 부분은 국지적으로 미세한 움직임에 한하여 발생한다.
- 2) Motion Missing 에러는 움직이는 영역과 정지 영역의 경계 부분에서 발생한다.

이를 바탕으로 먼저 입력 영상을 움직임이 있는 영역, 배경 영역, 경계 영역으로 구분하여 각각에 사용하는 임계 값을 달리 하여 정확성과 연산 효율을 높인다. 움직임이 있는 영역은 공간 필터를 사용하므로 3 단계 탐색을 수행할 필요가 없으므로 높은 임계 값을 주어 간단한 연산을 행하며 정지 영역의 경우 Motion Missing이 발생할 확률이 거의 없으므로 비교적 높은 임계 값을 주어 보간 할 화소 값을 결정한다. 하지만 경계 부분의 경우 Motion Missing이 일어날 수 있으므로 낮은

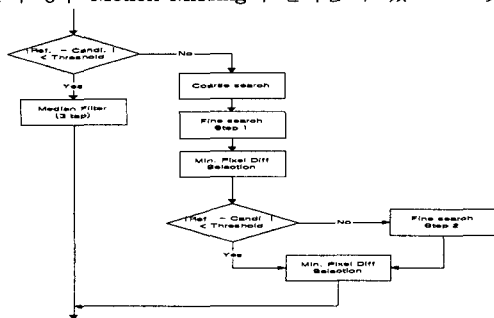


그림 6. 3 단계 최적 화소 탐색

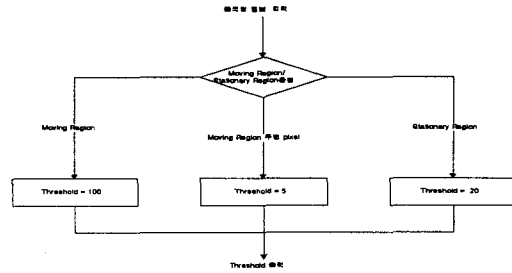


그림 7. 임계 값의 결정

임계 값을 주어 3 단계에 걸쳐 탐색을 수행하여 정확한 보간이 이루어지게 한다. 그림 6은 3 단계 최적 화소 값을 탐색하는 흐름도이고 그림 7은 임계 값을 결정하는 흐름도이다. 여기에 사용되는 임계 값은 모의 실험을 통해 얻은 값이다.

5. Switch 블록

위의 방법을 통해서 구해진 화소 값에 대해 AMPD Filter를 사용하여 보간한 값을 최종 화소 값으로 사용할 지, Edge based Median Filter를 이용하여 보간한 화소 값을 사용할 지, 그 두 값을 혼용하여 사용할 지를 결정하는 블록이다. 그림 8은 이를 결정하는 계수 Alpha를 결정하는 식과 최종적으로 보간 되는 'z' 화소 값을 결정하는 식이다. 여기서 'Med'는 Maximum Detection 블록에서 구해진 최대 움직임 값에 대해 노이즈 제거를 위해 Median Filter를 통과한 값을 나타내며 H와 L은 상한 값과 하한 값을 각각 가리킨다. 여기에 사용되는 상, 하한 임계 값은 영상의 종류에 따라 다르다. 앞서 임계값 결정 블록에서 설명한 것처럼 입력 영상을 움직임이 있는 영상, 보통 영상, 움직임이 없는 영상, 정지 영상으로 구분하고 각각에 대해 다른 임계 값을 결정하여 성능을 향상시킨다.

$$\alpha = \begin{cases} 0 & , Med^k \leq Th_L \\ \frac{Med^k - Th_L}{Th_H - Th_L} & , Med^k \leq Th_H \\ 1 & , Med^k \geq Th_H \end{cases}$$

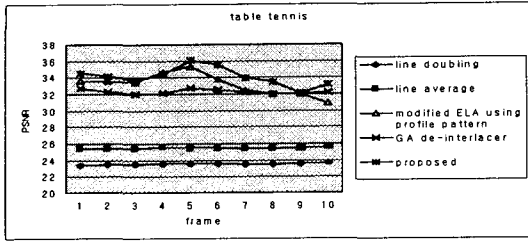
$$z = \alpha * AMPD(z) + (1 - \alpha) * EBMED(z)$$

그림 8. 최종 화소 값의 결정

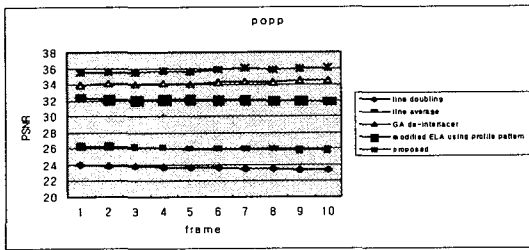
III. 모의 실험 및 결과

Motion Adaptive 방법 중에서 중요한 부분은 False Motion, Motion Missing, Moving Diagonal Edge의 처리이다. 높은 PSNR을 가진다고 하더라도 특히 Motion Missing 에러가 발생할 경우 급격한 화질의 열화를 가져오므로 정확한 성능 분석을 위해 PSNR의 주관적인 화질 평가를 병행하였다. 정확한 PSNR을 구하기 위해서 352*240 progressive 영상을 사용하였으며 주관적인 화질 평가를 위해서 704*480 Interlace 영상을 사용하였다. 공통적으로 사용되는 test 영상은 Line Doubling, Line Average, ELA(Edge-based Line Average), GA HDTV scan converter와 Brightness Profile Pattern을 이용한 ELA방법(Modified ELA) 등을

같이 실험하였다[2][3][6][7]. 사용된 영상은 Table Tennis, Mobile, Susi, Flower, Football, Popp 등이다.



(a) 352*240 popp image



(b) 352*240 table tennis image
그림 9. 입력 영상에 따른 PSNR

method \ image	Susi	table Tennis	Football	Mobile	Popp
Line Doubling	30.198	23.503	25.529	17.922	23.52
Line Average	34.465	25.457	29.345	20.05	26.00
GA HDTV de-interlacer	40.363	33.142	33.584	28.76	35.424
Modified ELA using Profile Pattern	35.09	32.279	34.887	29.236	32.986
Proposed	41.262	34.282	34.882	29.138	35.638

그림 10. 입력 영상에 따른 평균 PSNR

그림 9는 입력 영상에 따른 PSNR이며 그림 10은 각각의 영상에 따른 평균 PSNR을 나타내었다. 제안방법은 각각의 방법에 비해 평균적으로 우수한 PSNR을 보였다. Edge 성분이 많은 입력 영상에 대해서는 Modified ELA방법이 GA de-interlacer에 비해 Edge 처리에 대한 성능이 뛰어나므로 높은 PSNR을 보였으며 일반적인 영상에서는 Median Filter를 사용하는 GA de-interlacer가 높은 PSNR을 보였다. 제안 방법의 경우 공간 필터로 Edge based Median Filter를 사용하고 Temporal Filter로 AMPD Filter를 사용하기 때문에 모든 입력 영상에서 대체로 높은 PSNR을 보였다. 그러나 De-interlacer의 성능을 단순한 PSNR로 측정하기는 어렵다. 이는 상대적으로 높은 PSNR을 가지더라도 Motion Missing이나 False Motion이 발생할 경우 급격한 화질의 열화를 발생하기 때문이다. 이러한 화질의 열화는 그림 11처럼 GA De-interlacer에서 많이 발생한다. 그림 12는 각 방법으로 보간된 영상이다.

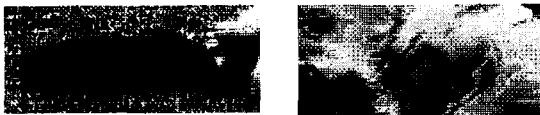


그림 11. False Interpolation for fast image region

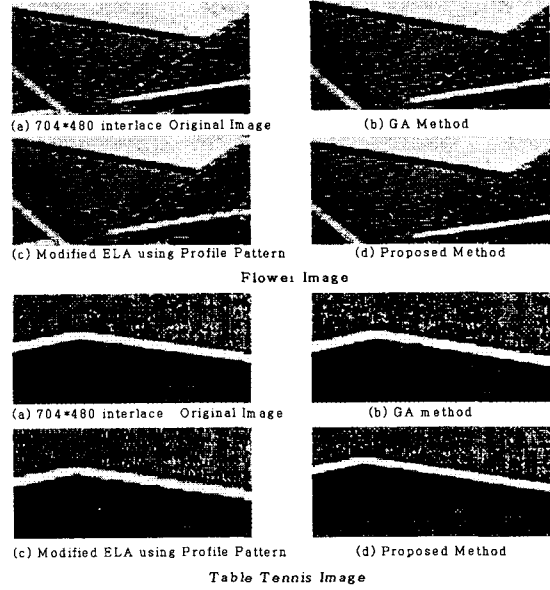


그림 12. Interpolation(Table Tennis Image)

IV. 결론

본 논문에서는 효율적인 보간 알고리즘을 제안했다. 공간 필터로서 Moving Diagonal Edge와 False Motion의 처리를 위해 Edge based Median Filter를 사용하여 효율적인 보간이 이루어지게 하였으며 Motion Missing 에러를 제거하기 위해 입력 영상을 4가지 유형으로 분류하고 다시 움직임이 있는 영역, 배경 영역, 경계 영역으로 나누어 각각 다른 임계 값을 설정하여 3 단계에 걸쳐 정확한 화소 값을 결정하는 AMPD Filter를 제안했다. 모의 실험 결과 제안 알고리즘은 기존의 Motion Adaptive 방법에 나타나는 문제점을 모두 제거하였으며 기존의 알고리즘에 비해 뛰어난 성능을 발휘했다.

V. Reference

- [1] Gerard De Hann and Erwin B. Bellers, De-interlacing - An Overview, Proceedings of IEEE, Vol. 86, No. 9, September 1998
- [2] Tao Chen, Hong Ren Wu, Zheng Hua Yu, An Efficient Edge Line Average Interpolation Algorithm for De-interlacing, In Visual Communications and Image Processing 2000, Proceedings of SPIE Vol. 4067(2000)
- [3] C.J.Kuo, C. Liao, and C.C.Lin, Adaptive Interpolation Technique for scanning rate conversion, IEEE Trans, Circuits Syst. Video Technol 6(3), pp317 -321, 1996
- [4] D. Hargreaves and J. Vaisey, Bayesian motion estimation and interpolation in the interlaced video sequence, IEEE Trans. Image Processing 6(5),pp764-769, 1997
- [5] Renxiang Li, Bing Zeng and Ming L.Liou, Reliable Motion Detection/Compensation for Interlaced Sequence and Its Applications to De-interlacing, IEEE Trans. on Circuit and Systems for Video Technology, Vol. 10, No. 1, Feb. 2000
- [6] B. Bhatt, F.Templin, B.Hogstrom, H.Derovanessian, S.Lamadrid, and J. Mailhot, Grand Alliance HDTV multi-format scan converter, IEEE Trans. Consumer Electron, Vol. 41, pp1020-1031, Nov. 1995
- [7] Dongil Han, Chang-Yong Shin, Seung-Jong Choi and Jong-Seok Park, A Motion Adaptive 3-D De-interlacing algorithm based on the brightness profile pattern difference, IEEE Trans. Consumer Electron, Vol. 45, No. 3, August 1999