

움직임 보상을 통한 움직임 기반의

De-interlacing 기법

이성규 , 강석규 , 이동호
한양대학교 전자 전기 제어계측 공학과

A Motion-Adaptive De-interlacing Method using Motion Compensated Interpolation

Sung-Kyu Lee and Suk-Kyu Kang and Dong-Ho Lee

Department of Electronic, Electrical, Control and Instrumentation Engineering of Hanyang University

E-mail: kkang@image.hayang.ac.kr dhlee@image.hayang.ac.kr

요 약

본 논문에서는 움직임 보상을 이용한 Motion-Adaptive De-interlacing Method를 제안 한다. 정확한 움직임 추정을 위해서 Pre-filter로서 EBMF(Edge Based Median Filter)를 사용하며 새로운 Block Matching Method를 제안한다. Temporal Filter로서 Motion Missing Error를 제거하기 위해 입력 영상의 움직임 영역에 따라 각각 다른 임계 값을 적용하는 AMPDF(Adaptive Minimum Pixel Difference Filter)를 적용하였으며 MMD(Maximum Motion Detection)와 SAD(Sum of Difference)를 이용하여 빠른 움직임 영역에서 화질을 향상시켰다. 최종적으로 잘못된 움직임 보상에 기인하는 화질의 열화를 방지하기 위한 Motion Correction Filter를 제안한다.

No-MC Method에는 다양한 공간 필터를 사용한 접근 방법과 움직임에 따라 적응적으로 변하는(Motion Adaptive) 방법 등이 있다[1]. 최근에는 비교적 많은 연산량이 요구되지만 정확한 보간(Interpolation)을 위해 움직임 보상을 이용한 Motion Adaptive De-interlacing Method가 많이 사용된다. 그러나 이러한 접근 방법은 정확한 움직임 추정을 전제로 한다. 일반적으로 움직임 추정을 위한 전단계로 Line Average를 pre-filter로 사용하는데 이는 구현은 용이하지만 정확한 움직임 추정을 보장할 수 없고 특히 Edge 부분에서 화질의 열화를 가져온다[1][4]. 본 논문에서는 이러한 문제점을 해결하기 위해서 Pre-filter로서 Edge에 중점을 두어 보간하는 EBMF(Edge based Median Filter)를 제안하여 Edge에 대한 성능을 향상시켰으며 De-interlacing에 적당한 새로운 움직임 추정 기법을 제안한다.

I. INTRODUCTION

현재의 TV System은 Interlace Scan에 기인하는 Edge Flicker와 Interline Flicker, Line Crawling등이 발생하는 단점이 있으며 이러한 단점을 보완하기 위해 De-interlacing이 널리 사용되고 있다. 지금까지 다양한 De-interlacing Method가 제시 되었는데 이는 크게 MC(Motion Compensation) Method와 No-MC(No Motion Compensation) Method로 나눌 수 있으며

II. PROPOSED DE-INTERLACING LGORITHM

그림 1은 제안한 De-interlacer의 블록도이다. Pre-Processing Block에서는 먼저 정확한 움직임 영역을 추출하기 위해서 별도의 움직임 감지 블록을 두어 Motion Vector 값과 함께 정확한 움직임 영역을 감지한다.

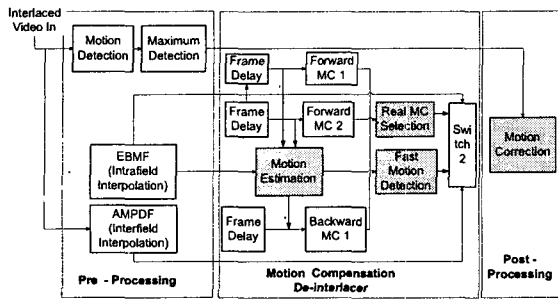


그림 1. Block Diagram using Motion Compensated Interpolation

참조 영상에서 존재하는 라인 사이의 값을 보간하기 위해 EBMF(Edged Based Median Filter)를 통과하여 프레임 구조로 변환하고 AMPDF(Adaptive Minimum Pixel Difference Filter)를 사용하여 최종 Switch블록에서 정지 영역에서 사용될 보간 값을 구한다. Motion Compensated De-interlacer Block에서는 3개의 참조 영상으로부터 움직임 보상을 행한 후 Real MC Switch Block에서는 이 중 EBMF로부터 보간된 값은 버리고 원 영상에 해당되는 화소 값만을 보간에 사용한다. 또한 빠른 움직임 영역을 감지하기 위해서 MMD(Maximum Motion Detection) 값과 SAD(Sum of Absolute Difference)값을 사용하여 빠른 움직임 영역을 정한다. Motion Compensated De-interlacer에 사용된 움직임 보상은 구현의 용이성을 위해서 Block Matching Algorithm을 사용하게 되는데 Block별로 SAD를 구하여 빠른 움직임 영역을 판별할 경우 다른 블록에 걸쳐있는 영역에 대해서는 감지가 불가능하다. 그러므로 이러한 블록에서 빠른 움직임이 존재하는 영역의 화질의 열화를 막기 위해서 수직 축으로 Median Filter를 통과시킨다. 각각을 블록별로 설명하면 다음과 같다.

1. EBMF(Edge Based Median Filter)

Pre-Filter로 기존방법들의 문제점을 보완하기 EBMF(Edge Based Median Filter)를 사용한다. 이는 그림 2와 같이 2 단계로 나누어지는데 1차적으로 3x3 Sobel Operator를 사용하여 Edge Map를 추출하고 7x3 Mask를 사용하여 완만한 Edge에 대한 성능을 향상시켰다.

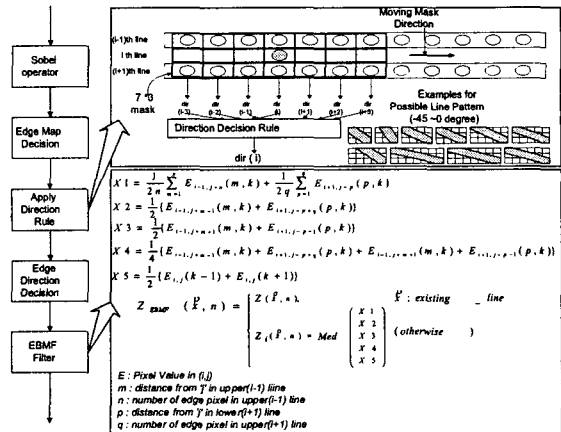


그림 2: EBMF(Edge Based Median Filter)

2. AMPDF(Adaptive Minimum Pixel Difference Filter)

먼저 입력 영상을 움직임이 있는 영역, 배경 영역, 경계 영역으로 구분하여 움직임이 있는 영역의 경우 움직임 보상된 값을 이용하므로 높은 임계 값을 주어 간단한 연산을 행하며 정지 영역의 경우 Motion Missing이 발생할 확률이 없으므로 비교적 높은 임계 값을 주어 보간할 화소 값을 결정 하지만 경계 영역의 경우 Motion Missing이 발생할 수 있으므로 임계 값을 낮게 주어 3 단계 탐색을 수행하여 정확한 보간이 이루어지게 한다.

3. Motion Compensated Interpolation

그림3은 제안하는 Motion Compensated Interpolation의 블록도이다. 먼저 양방향 움직임 추정을 통해서 각각의 Motion Vector 값을 구하고 Motion Vector Refinement Block을 두어 원 영상에 대한 Motion Vector 값만을 움직임 보상에 이용한다. 마지막 단계로 원 영상에 대한 실제 Motion Vector 값에 대해 움직임 보상된 각 블록의 평균값을 이용하여 최종적으로 보간한다.

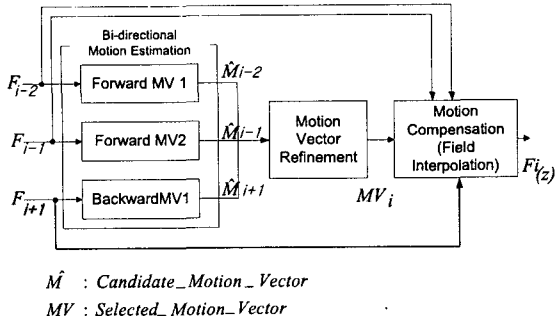


그림 3. Proposed Motion Compensated Interpolation

Motion Estimation

본 논문에서는 연산의 효율성과 구현의 용이성을 위해서 블록 정합 방법을 사용하며 블록 정합 기준으로는 식 (1)과 같이 SAD(Sum of Absolute Difference)를 이용한다. 정확한 움직임 추정을 위해서 먼저 EBMF를 통해 Missing Line을 보간하고 움직임 추정을 행하게 된다.

$$SAD(i, j) = \frac{1}{mn} \sum_k \sum_l |S_f(k, l) - S_{f-1}(k+i, l+j)|$$

$$MIN = \min |SAD(i, j)|$$

$$motion_vector = (i, j) \text{ with } MIN_value$$

(1)

where, S : pixel_Value

f : field_number

$k \in \{0 \leq k \leq 7\}$, horizontal_direction

$l \in \{1, 3, 5, 7\}$, vertical_direction

i, j : in_search_range

Motion Vector Refinement

구해진 Motion Vector를 사용하여 실제 움직임 보상에 사용될 Motion Vector를 구한다. Motion Vector의 경우 보간될 위치의 라인이 원 영상일수도 있고 EBMF를 통해 보간된 영상일 수도 있다. EBMF를 이용할 경우 움직임 보상의 의미가 없으므로 원 영상에 대한 Motion Vector만을 이용하기 위해서 그림 4와 같이 원 영상을 이용할 수 있는 Real Motion Vector를 구한다.

Motion Compensation (Field Interpolation)

그림 5는 3 개의 참조 영상에서 얻어진 Motion Vector를 바탕으로 실제로 움직임 보상에 이용되는 움직임 보상 값을 구하는 흐름도이다. Real Motion Vector를 가지는 움직임 보상 값을 최종 움직임 보

상에 이용한다. Real Motion Vector가 없을 경우에는 움직임 보상 값을 그대로 사용하면 화질의 급격한 열화를 가져오므로 이러한 부분에 대해서는 EBMF 값을 사용하여 보간 한다.

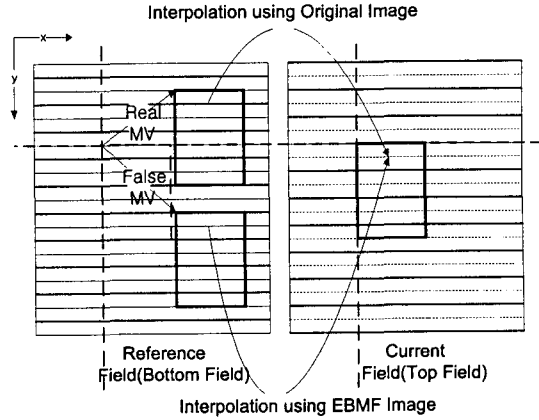


그림 4. Real Motion Vector Selection

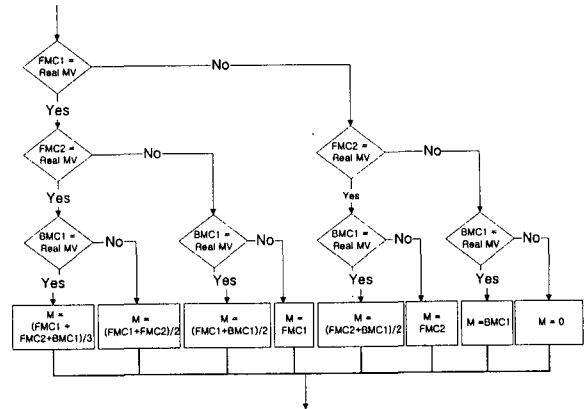


그림 5. Flow Chart for Real Motion Compensation

Fast Motion Detection & Interpolation

제안한 De-interlacer는 블록단위로 MMV(Maximum Motion Value)와 SAD(Sum of Absolute Difference)를 이용하여 빠른 움직임을 감지 한다. SAD 값이 크게 나타나는 블록의 경우 Mismatching이 일어나는 블록으로 가정하여 MMV와 비교하고 최종적으로 빠른 움직임 영역을 탐색한다.

Pel-Based Motion-Adaptive Interpolation (Switch Block)

그림 6은 최종 화소 값을 보간하기 위한 흐름도이다. 입력 영상에서 배경에 대해서는 AMPDF를 사용하여

보간 한다. 일반적인 영역에서는 상,하한 임계값을 두어 상한 임계 값 이상일 경우 움직임 보상된 값을 사용하며 하한 임계 값과 사이일 경우 움직임 보상된 값과 AMPDF값을 혼용하여 사용한다. 하한 임계 값 이하일 경우 순수한 AMPDF 값만을 사용하며 빠른 움직임 영역의 경우 EBMF값을 그대로 사용하여 화질의 열화를 방지한다.

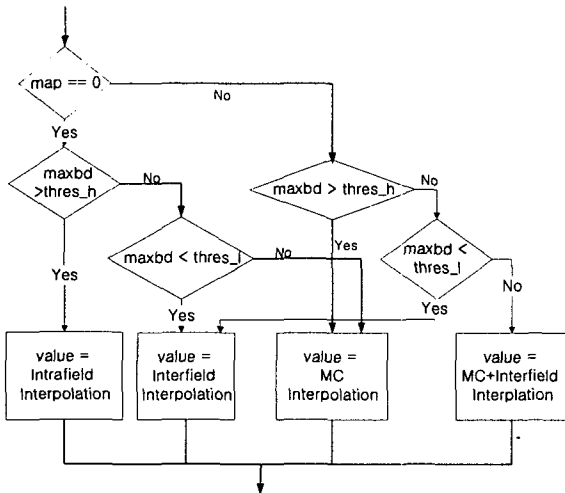


그림 6. Flow Chart for Switch Block

최종적으로 보간되는 'z'의 값은 식(2)와 같다.

$$z = \beta * (\alpha * mc(z) + (1-\alpha) * temp(z)) + (1-\beta) * spat(z)$$

$mc(z)$: motion_compensate d_Interpolation_in_'z'
 $temp(z)$: AMPDF_in_'z'
 $spat(z)$: EBMF_in_'z'

$$\alpha = \begin{cases} 0 & , MC_i(z) \leq Th_L \\ \frac{MC_i(z) - Th_L}{Th_H - Th_L} & , Th_L \leq MC_i(z) \leq Th_H \\ 1 & , MC_i(z) \geq Th_H \end{cases}$$

Th_H : upper_threshold_value
 Th_L : lower_threshold_value

III. SIMULATION RESULT

표2는 각각의 방법에 대한 평균 PSNR을 나타낸다. Ref. 1[2]의 경우 비교적 구현은 간단하지만 PSNR이 전반적으로 떨어졌으며 Ref. 2[3]의 경우 Block Effect는 제거되어 지고 No MC Method에 비해 3~6dB 정도의 성능 향상은 있었지만 전체적으로 Smoothing Filter를 사용함으로써 전반적으로 화면이 무더지는 단점이 나타난다. 제안 방법은 움직임 감지 블록을 따로 두고 각각의 움직임에 따라 각각 다른 방법을 적용하여 보간 하

기 때문에 빠른 움직임 영역에서도 화질의 열화가 없으며 Motion Correction Block을 따로 두어 Block Effect 현상을 제거하였기 때문에 모든 영상에서 비교적 높은 PSNR을 얻을 수 있었다.

표 1. Test Image Characteristic

Test 영상	영상의 분류
Table Tennis	부분적으로 변화 하는 영상 (공 주변 빠른 움직임 변화)
Flower Garden	화면이 전체적으로 움직이는 영상 (좌->우) Moving Diagonal Edge 존재
Football	전체적으로 빠른 움직임 변화가 있는 영상
Susie	움직임이 거의 없고 부분적으로 변화하는 영상
Popp	부분적으로 변화하는 영상. 가느 Edge line이 많이 존재

표 2. 입력 영상에 따른 평균 PSNR

Image Method	Susie	Table Tennis	Football	Popp
Conventional MCD Phase filter사용 [1]	33.817	25.362	26.258	25.081
Ref. 1 [2]	38.545	28.167	30.102	28.045
Ref. 2 [3]	39.468	32.457	33.345	33.010
Proposed Method	42.848	35.845	34.595	36.838

REFERENCES

- [1] Gerard De Haan and Erwin B. Bellers, "De-interlacing - An Overview", Proceedings of IEEE, Vol. 86, No. 9, pp.1839 - 1857, September 1998
- [2] Renxiang Li, Bing Zeng "Reliable Motion Detection/Compensation for Interlaced Sequence and Its Applications to Deinterlacing", IEEE Trans. on Circuit and Systems for Video Technology, Vol. 10, No. 1, pp. 23-29, Feb. 2000
- [3] K.J.Sugiyama and H.Y.Nakamura, "A Method of De-interlacing with Motion Compensated Interpolation", IEEE Trans. Consumer Electronics, Vol. 45, No. 3, pp.611-616, August 1999
- [4] F.M.Wang, D. Anastassiou, and A.N.Netravali, "Time-recursive de-interlacing for IDTV and pyramid coding", Signal Process, Image Commun. 2, pp365-374, 1990