

# 시공간 정보를 이용한 움직임 기반의 De-interlacing 기법

심 세 훈, 김 용 하, 정 제창  
한양대학교 전자통신공학과

## A Motion-Adaptive De-interlacing Method using Temporal and Spatial Domain Information

Sehun Shim, Yongha Kim, Jechang Jeong  
Department of Electronic Communications Engineering, Hanyang University  
E-mail : [ssh@ece.hanyang.ac.kr](mailto:ssh@ece.hanyang.ac.kr)

### Abstract

In this paper, we propose an efficient de-interlacing algorithm using temporal and spatial domain information. In the proposed scheme, motion estimation is performed same parity fields, i.e., if current field is even field, reference fields are previous even field and forward even field. And then motion vector refinement is performed to improve the accuracy of motion vectors. In the interpolating step, we use median filter to reduce the interpolation error caused by incorrect motion vector. Simulations conducted for various video sequences have shown the efficiency of the proposed interpolator with significant improvement over previous methods in terms of both PSNR and perceived image quality.

### I. 서론

현재 아날로그 TV 신호는 밴드 폭의 한계 때문에 Even, Odd field가 1/60초의 시간차이를 두고 교차되어 1frame을 구성하는 Interlaced signal을 사용하고 있다. 1장의 frame을 구성하는 even, odd field는 서로 다른 시간(1/60초의 차이)대에 존재하기 때문에 아날로그 TV의 영상을 DTV나 Video printer, Computer에서 그대로 사용한다면 움직임이 있는 부분에서 심각한

artifacts가 발생한다. 이러한 문제점들을 극복하기 위해 지금까지 다양한 De-interlacing 방법들이 제시되었으며 이는 크게 여러 frames의 정보를 사용하여 값을 보간하는 MC(Motion Compensation) 방법과 1장의 frame 정보를 사용하여 값을 보간하는 NO-MC(No Motion Compensation) 방법으로 나눌 수 있다.

본 논문에서는 MC와 NO-MC의 결과를 Median Filter를 사용하여 스위칭함으로써 급격한 화질의 열화를 방지할 수 있는 효율적인 De-interlacing 방법을 제시하였다.

### II. 기존의 De-interlacing 알고리즘

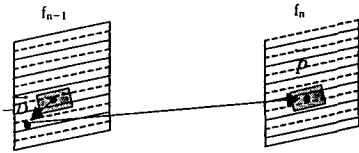
#### 2.1 No-MC(No Motion Compensation)

기존의 No-MC De-interlacing 방법 중에는 ZOI(Zero Order Interpolation), FOI(First Order Interpolation), ELA(Edge Based Line Average Interpolation), E-ELA(Efficient Edge Line Average Interpolation) 방식 등이 있다.

ELA알고리즘은 작은 픽셀 값의 변화에 민감하고 45도보다 크거나 작은 각도의 대각선에 대해서는 잘못된 Edge 정보를 이용하는 등의 단점이 있다. 그래서 E-ELA알고리즘은 ELA보다 더 정확한 방향성을 찾기 위해 현재 위치에서 영상의 공간적인 형태를 세가지 범주로 분류한다. 공간적인 방향성이 왼쪽, 오른쪽, 또는 그 외의 경우로 분류하여 각 분류된 범주에서 적절한 방법으로 ELA를 변형하여 적용한다[3][4].

### 2.2 MC (Motion Compensation)

MC를 이용한 De-interlacing 방법은 일반적으로 ELA 알고리즘을 응용한 방법들보다 좋은 결과를 얻을 수 있다.



[그림 1] Motion Compensation

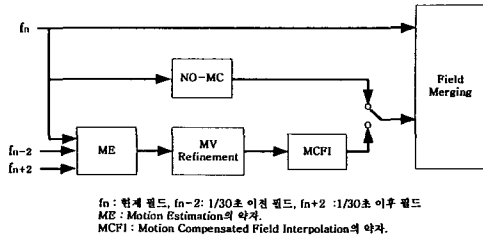
MC의 기본 개념은 [그림 1]에서 보는 바와 같다[2] [4] [5]. 새로이 채워질 픽셀  $F_n(\vec{p})$ 는  $f_{n-1}(\vec{p}-\vec{D})$ 로 정의되어지나 잘못된 Motion Vector를 찾거나 Blocking현상이 일어나므로 일반적으로 다음과 같이 정의되어진다.

$$F_n^r(\vec{p}) = \begin{cases} f_{n-1}^r(\vec{p}-\vec{D}) & \vec{p} \text{가 존재하는 Line일때,} \\ \text{Median} ( f_{n-1}^r(\vec{p}+\vec{u}), f_{n-1}^r(\vec{p}-\vec{u}), f_{n-1}^r(\vec{p}-\vec{D}) ) & \text{otherwise.} \end{cases}$$

$\vec{u}$ : 수직방향의단위벡터 (0,1)<sup>T</sup>

### III. 제안된 De-interlacing 알고리즘

#### 3.1 제안된 알고리즘의 전체 개요



$f_n$ : 현재 필드,  $f_{n-2}$ : 1/30초 이전 필드,  $f_{n+2}$ : 1/30초 이후 필드  
ME: Motion Estimation의 약자.  
MCPI: Motion Compensated Field Interpolation의 약자.

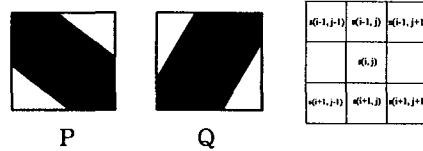
[그림 2] 제안된 De-interlacer의 블록 다이어그램

제안한 De-interlacer는 크게 NO-MC와 MC방법으로 보간할 값을 각각 정한 후 Median Filter를 사용하여 스 위칭함으로써 급격한 화질 열화의 발생을 방지한다. NO-MC에서는 [1]에서 제안한 알고리즘의 방향성 검사를 위한 윈도우 크기를 3×3으로 고정하여 사용하며, MC에서는 기존의 알고리즘에서 1/60초 이전의 필드에서 움직임벡터를 찾음으로 인해 생기는 에러(모션이 없는 부분도 있는 것으로 판별함)를 방지하기 위해서 1/30초 이전의 같은 종류의 필드에서 움직임 예측을 수행하고, 보다 정확한 움직임 벡터를 얻기 위해 이를 MV refinement 과정을 통해 세부 조정을 실시한다.

#### 3.2 NO-MC (No Motion Compensation)

고전적인 ELA 알고리즘은 간단한 계산과 구현의 용이성으로 인해 널리 사용되고 있으나, 45도보다 크거나

작은 각도의 대각선에 대해서는 잘못된 Edge 정보를 이용하는 단점이 있으므로 이의 개선과 구현의 용이성을 위해 [1]에서 제안한 E-ELA 알고리즘을 변형하여 사용한다. 제안한 알고리즘의 방향성 검사를 위한 윈도우 크기는 3×3으로 고정하여 사용한다.



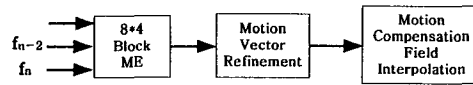
3×3 블록 내에서 Edge가 가질 수 있는 방향성을 P와 Q로 구분하고 P와 Q를 다음과 같이 정의 한다.

$$P = |s(i-1, j) - s(i+1, j+1)| + |s(i-1, j-1) - s(i+1, j)|$$

$$Q = |s(i-1, j+1) - s(i+1, j)| + |s(i-1, j) - s(i+1, j-1)|$$

위의 P, Q를 이용하여 3×3윈도우 내에서 P>Q, P=Q, P<Q로 나누어 방향성을 유추하고 상관도 검사를 통해 적절한 값으로 보간한다.

#### 3.3 MCFI (Motion Compensated Field Interpolation)

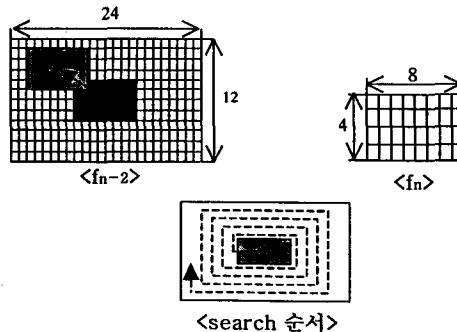


[그림 3] 제안한 ME 및 MCFI

현재 field  $f_n$ 를 기준으로 1/30초 이전 필드  $f_{n-2}$ 와  $f_{n+2}$ 에서 탐색 범위(search range) 24×12, 블록 8×4의 크기로 Motion을 1차 탐색한다.

1차에서 구한 Motion Vector를 기준으로 8×4의 블록을 search range로 하고 4×2 크기의 블록으로 Motion Vector를 재조정한다.

##### 3.3.1 MC (Motion Compensation)



[그림 4] ME의 전체 개요 및 search순서

$f_n$ 의 8×4블록의 위치를 기준으로  $f_{n-2}$ 과  $f_{n+2}$ 에서 위 그림과 같은 search 순서로 탐색하여 Motion Vector

$f_{forward}, \vec{D}_{backward}$  를 구한다.

Motion Vector를 양쪽 방향으로 찾은 이유는 대부분의 화질의 열화가 잘못된 모션을 찾는 데 기인하기 때문에 Motion Vector의 신뢰도를 높이기 위해서 이다.

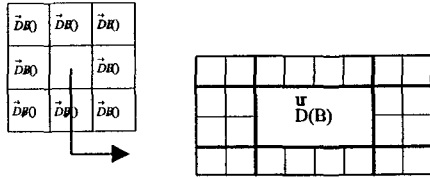
모션을 찾는 데 참조 필드로 1/30초 이전과 이후의 필드를 선택한 이유는 기존의 De-interlacing 방법들이 1/60초 이전의 다른 종류의 필드에서 모션을 찾음으로 인해 모션이 없음에도 불구하고 있는 것으로 잘못 찾는 경우가 발생하였기에 이를 개선하기 위해서 이다.

### 3.3.2 Motion Vector Refinement

3.3.1에서 구한 Motion Vector  $\vec{D}_{backward}, \vec{D}_{forward}$ 의 평균값인  $\vec{D}_{average}$ 을 기준으로  $f_n$ 를 기준필드,  $f_{n-1}$ 를 참조필드로 하여 Motion Vector를 재조정 한다.

$$(\vec{D}_{average} = \text{round}[(\vec{D}_{backward} + \vec{D}_{forward})/2])$$

이때 구한 Motion Vector를  $\vec{D}(B)$ 라 하고, [그림 5]과 같이 새로운 Motion Vector를 정의한다.



[그림 5] Motion Vector Refinement

중앙 4×2 블록 대해서는 해당 8×4 블록 Motion Vector를 적용하고 나머지 부분은 인접한 블록의 Motion Vector와의 평균값을 해당 MV로 취한다. 이는 블록 내의 모든 픽셀들에 동일한 MV를 적용함으로써 인해 생기는 화질의 열화를 방지하기 위함이다. 인접한 블록 사이에 MV의 차이가 큰 경우 평균을 취함으로써 인해 잘못된 MV를 취하게 되나 이렇게 잘못 구해진 MV 값은 마지막 보간 단계에서 median 필터를 사용하여 걸러지므로 결과 이미지는 영향을 미치지 않는다.

### 3.3.3 MCFI(Motion Compensated Field Interpolation)

제안한 알고리즘의 핵심은 MV가 없는 부분에 대해서 정확히 판별하는 것과 잘못된 MV로 인한 급격한 화질의 열화를 방지하는데 있다. 따라서 새로이 채워질 픽셀의 MCFI결과는 다음과 같이 정의된다.

$$f_{inter}(p) = \begin{cases} \vec{u} & \text{if } \vec{D}_{backward} = 0, & f_{n-1}(p) \\ \vec{u} & \text{else if } \vec{D}_{forward} = 0, & f_{n+1}(p) \\ \text{round}[(f_{n-1}(p) + f_{n+1}(p))/2] & \text{else if } \vec{D}_{backward} = 0 \text{ and } \vec{D}_{forward} = 0, \\ \text{round}[(f_{n-1}(p - \vec{D}(B)) + f_{n+1}(p + \vec{D}(B)))/2] & \text{else} \end{cases}$$

### 3.4 Field Merging

$$F_n(\vec{u}) = \begin{cases} f_n(\vec{u}) : \vec{u} \text{가 존재하는 라인일 때,} \\ \text{Median}(A, B, C) : \text{otherwise.} \end{cases}$$

A, B, C는 다음과 같이 정의된다.

A =  $f_{inter}(\vec{u})$ , MC의 결과

B = 3.2절 No\_MC의 결과  $f_{inter}(x, y)$ ,

C =  $\text{round}[\{f(x, y-1) + f(x, y+1)\}/2]$

## IV. 실험결과

제안한 방법의 성능 분석을 위하여 352×240 progressive영상을 352×120필드 구조의 interlaced 영상으로 변환하여 각 알고리즘을 적용하여 보간한 후에 PSNR과 주관적 화질을 비교하였다. 성능 비교를 위해 line doubling, line averaging, conventional MCD 등을 같이 실험하였다.

표 1. 각 영상의 Frame 별 PSNR [dB]

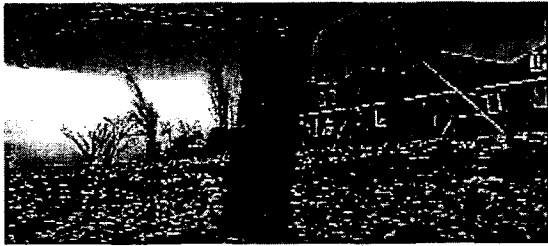
| Frame number | Flower garden | Table tennis | Football |
|--------------|---------------|--------------|----------|
| 1            | 27.49         | 33.03        | 31.79    |
| 2            | 27.68         | 32.97        | 31.55    |
| 3            | 26.85         | 33.09        | 30.54    |
| 4            | 26.12         | 32.83        | 30.01    |
| 5            | 26.56         | 32.64        | 30.47    |
| 6            | 27.54         | 32.90        | 30.23    |
| 7            | 28.14         | 32.99        | 29.22    |
| 8            | 28.04         | 33.15        | 29.06    |
| 9            | 27.69         | 33.09        | 29.66    |
| 10           | 25.98         | 33.51        | 29.37    |

표 2. 각 영상의 방법별 평균 PSNR 비교 [dB]

|                  | Flower garden | Table tennis | Football |
|------------------|---------------|--------------|----------|
| Line Doubling    | 17.53         | 23.50        | 25.53    |
| Line Averaging   | 19.21         | 25.46        | 28.35    |
| Conventional MCD | 17.49         | 30.08        | 26.26    |
| Proposed Method  | 27.21         | 33.02        | 30.19    |

실험에 사용된 영상은 352x240 Table tennis, Flower garden, Football 영상이다.

표 2와 [그림 6]에서 제안한 방법은 기존의 방법보다 PSNR과 화질에서 모두 우수한 결과를 보였다. 특히 카메라가 panning 하고 전체적으로 섬세한 edge가 많은 flower garden에서 매우 우수한 결과를 보여주었다.



(a) Line Doubling



(b) Line Averaging



(c) Conventional MCD



(d) Proposed Method

[그림 6] Flower Garden 영상의 deinterlacing 결과

## V. 결론

많은 motion adaptive de-interlacing 방법들에서 중요한 부분은 정확한 움직임을 찾아서 보간하는 것이다. 그러나 기존의 알고리즘은 1/60초 이전의 필드를 사용하였기 때문에 움직임이 없는 부분도 있는 것으로 잘못 인식하여 전체적인 화질의 열화를 가져왔다

본 논문에서는 이런 단점을 극복하기 위해 motion vector 를 2 단계로 나누어서 찾았다. 1 단계는 1/30 초 이전의 필드에서 움직임을 찾았고, 2 단계에서는 1 단계에 찾은 motion 을 다시 재조정하여 정확한 motion 을 찾는데 주력하였다. 그 결과 움직임이 없는 배경이나, 전체적으로 일정한 움직임이 발생하는 영상에 있어서 기존의 방법보다 뛰어난 결과를 보여주었다.

## VI. 참고문헌

- [1] T. Chen, H.R. Wu and Z. Yu, "Efficient edge line average interpolation algorithm for de-interlacing," *Proceedings of SPIE, Visual Communications and Image Processing 2000*, vol. 4067, pp.1551-1558, Perth, Australia, June 2000.
- [2] Y.Y. Jung, B.T. Choi, Y.J. Park and S.J. Ko, "An Effective De-Interlacing Technique Using Motion Compensated Interpolation," *IEEE Trans. Consumer Electronics*, vol. 46, no.3, August. 2000.
- [3] M.H. Lee, J.H. Kim, J.S. Lee, K.K. Ryu, and D.I. Song, "A New Algorithm for Interlaced to Progressive Scan Conversion Based on Directional Correlations and its IC Design," *IEEE Trans. Consumer Electronics*, vol. 40, no.2, p p. 119-129, May. 1994.
- [4] Gerard De Hann and Erwin B.Bellers, "De-interlacing An Overview," *Proceedings of IEEE*, vol. 86, no. 9, pp.1839-1857, September 1998.
- [5] K.J. Sugiyama and H.Y. Nakamura, "A Method of Deinterlacing with Motion Compensated Interpolation", *IEEE Trans. Consumer Electronics*, vol. 45, no. 3, pp. 611-616, August 1999