

전 방향 에지 예측 기법을 이용한 De-interlacing 알고리즘

준희원 구수일*, 이세영*, 강근화*, 정희원 정제창*

Novel De-interlacing Algorithm Using All Direction Edges Estimation Technique

Suil Ku*, Se-young Lee*, Kunhwa Kang* *Associate Members*,
Jechang Jeong* *Regular Member*

요약

본 논문은 전 방향 에지예측 기법을 이용한 De-interlacing 알고리즘을 제안한다. 본 논문에서는 기존의 DOI(Direction-Oriented Interpolation) 알고리즘을 기본 방법으로 사용 하였다. De-interlacing 방법은 크게 2단계로 나누어진다. 먼저 에지의 방향을 예측한 후, 잃어버린 화소 값을 에지의 방향에 따라 보간하는 방법이다. 본 논문에서는 고각도 에지를 고려한 DOI 알고리즘을 통하여 에지를 예측한 후 잃어버린 화소값을 중간값(median) 필터를 사용하여 보간한다. 실험 결과는 제안된 알고리즘이 기존의 알고리즘들 보다 객관적 및 주관적인 평가에서 우수함을 입증한다.

Key Words : De-interlacing, Image interpolation, Edge Estimation, Median Filter, Post Processing

ABSTRACT

This paper proposed a novel de-interlacing algorithm using all direction edges estimation technique. In the proposed algorithm, previously developed the DOI(Direction-Oriented Interpolation) algorithm was used as a basis. The de-interlacing method was divided into two main parts. First, we should estimate edge direction. Then, missing pixels were interpolated along with the decided edge. In this paper, after estimating the edge through the DOI algorithm considering high angle edge direction, missing pixels were interpolated by using the median filter. Experimental results indicate that the proposed algorithm is superior to the conventional algorithms in terms of the objective and subjective criteria.

1. 서론

현재의 TV 디스플레이 시스템은 하드웨어의 제약으로 하나의 프레임에 대해 홀수 줄만을 갖는 홀수 필드와 짝수 줄만을 갖는 짝수 필드로 나누어 주사하는 격행 주사방식을 사용한다. 즉, 한 프레임의 영상을 홀

수라인 또는 짝수라인을 제거한 격행 주사방식으로 데이터의 양을 줄여서 전송 할 수 있다. 격행으로 주사된 데이터를 다시 순차주사로 바꾸기 위하여 잃어버린 줄을 다른 화소들을 참조하여 보간을 하게 되는데 이것을 De-interlacing라고 한다. 격행 주사방식을 사용하면 대역폭을 줄일 수 있는 장점이 있으나 주사 방식

※ 본 연구보고서는 정부통신부 출연금으로 ETRI, SoC 산업진흥센터에서 수행한 IT SoC 핵심설계인력양성사업의 연구결과입니다.

* 한양대학교 전자통신컴퓨터공학과 영상통신 및 신호처리 연구실(kusuil81@nate.com)

논문번호 : KICS2008-05-252, 접수일자 : 2008년 5월 29일, 최종논문접수일자 : 2008년 8월 29일

의 특징 때문에 화면사이 flicker 나 영상 에지에서의 crawling 혹은 twitter 와같이 원하지 않는 시각적 열화를 야기 할 수 있다. 이를 막기 위해 여러 가지 De-interlacing 알고리즘들이 제안되어 왔다.

기존의 De-interlacing[1-11] 방법들은 크게 2가지로 나뉜다. 움직임 정보를 이용하는 Inter-field De-interlacing[8-11] 방법과 움직임 정보를 사용하지 않는 Intra-field De-interlacing[2-7] 방법으로 분류 할 수 있다.

Inter-field De-interlacing 방법은 다시 움직임 적응 필터를 사용하여 접근하는 방법과 움직임 보상을 이용하는 방법으로 분류되어 진다. 이러한 Inter-field De-interlacing 방법은 일반적으로 움직임 정보가 신뢰성이 있다면 수요할 만한 결과를 이끌어 내지만 움직임 정보를 신뢰할 수 없다면 좋은 성능을 기대할 수 없을 뿐 아니라 하드웨어 구현 시 매우 복잡하다.

그리고 Intra-field De-interlacing 방법 또한 다시 두 가지 방법으로 분류할 수 있는데 다양한 공간 필터를 사용하여 접근하는 방법과 에지의 방향성을 고려하여 접근하는 방법이다. 영상의 에지와 그에 따른 방향성은 인간 시각에 매우 민감한 정보로서 이러한 정보를 고려하지 않고 De-interlacing을 수행하면 번짐 현상(blurring)을 초래하게 된다. 이러한 영상의 방향성을 고려한 Intra-field De-interlacing

방법 중 많이 사용하는 방법이 ELA(Edge-based Line Averaging) 알고리즘이다. ELA 알고리즘은 간단한 계산과 구현의 용이성으로 널리 사용되고 있으나 화소 값의 작은 변화에 민감하다는 단점이 있다. 이런 단점을 보완하기 위해 픽셀 단위로 에지의 방향성을 예측하는 것이 아니라 3x3블록 단위로 에지의 방향을 예측하게 되었다. 따라서 ELA 알고리즘보다는 블록 단위로 에지를 더욱 정확히 찾을 수 있는 DOI(Direction-Oriented Interpolation) 알고리즘을 기반으로 하였다.

DOI 알고리즘은 ELA 알고리즘보다 에지를 정확히 찾을 수 있지만 전 방향에 대하여 에지를 예측하지 못하는 단점을 가지고 있다. DOI 알고리즘 특성상 수직(90°)에지와 45° 이하의 저각도 에지에 대해서는 예측을 하지만 45° ~ 90° 사이의 고각도 에지는 예측을 하지 않는다.

본 논문에서는 DOI알고리즘에서 제안하는 수직(90°)에지와 45° 이하의 에지뿐만 아니라 45°-90° 사이의 고각도 에지도 고려하는 하는 전 방향 에지를 예측한다. 그리고 마지막으로 잃어버린 화소는 예측

된 후보 화소들을 중간값(median) 필터를 사용하여 보간 함으로써 영상의 contrast를 저해하는 blurring 효과나 impluse 잡음 등을 제거하여 효과적으로 보간 할 수 있다.

제 II장에서는 본 논문의 기본이 되는 기존의 알고리즘들에 대하여 살펴보고 제 III장에서는 제안한 기법을 이용한 De-interlacing 알고리즘에 대해 설명한다. 제 IV장에서는 제안하는 알고리즘의 실험 결과 및 분석을 하고, 마지막으로 제 V장에서 결론을 맺는다.

II. 기존 알고리즘

이번 장에서는 기존의 Intra-field De-interlacing 방법들 중 에지의 방향성을 고려한 알고리즘인 ELA 알고리즘과 이것보다는 더욱 향상된 E-ELA 알고리즘, 그리고 에지를 더 정확히 예측하는 DOI 알고리즘에 대해 살펴보도록 하겠다.

2.1 ELA (Edge-based Line Averaging) 알고리즘

Intra-field De-interlacing 알고리즘 중에서 가장 널리 사용되는 방법이 ELA 알고리즘이다.

ELA 알고리즘은 두 개의 인접 라인들 사이에 보간 라인을 선형적으로 보간하는데 픽셀들 간의 방향 상관을 사용한다. 에지 방향은 수직(vertical) 방향과, 대각선(diagonal-1, diagonal-2) 방향이 있을 수 있고, 위 3가지 방향 중에서 픽셀들 값의 차이가 최소인 방향을 에지 방향으로 선택한다. 그리고 에지 방향으로 선택된 두 화소의 평균값으로 보간한다.

$$\begin{cases} a = |X(i-1, j-1) - X(i+1, j+1)| \\ b = |X(i-1, j) - X(i+1, j)| \\ c = |X(i-1, j+1) - X(i+1, j-1)| \end{cases} \quad (1)$$

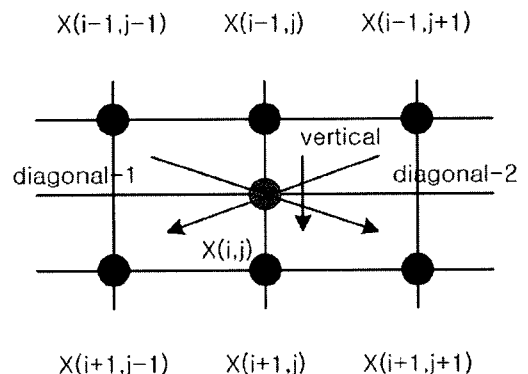


그림 1. ELA 알고리즘을 위한 3x3 윈도우

$$X(i,j) = \begin{cases} (X(i-1,j-1)+X(i+1,j+1))/2, & \text{if } \min(a,b,c)=a \\ (X(i-1,j)+X(i+1,j))/2, & \text{if } \min(a,b,c)=b \\ (X(i-1,j+1)+X(i+1,j-1))/2, & \text{if } \min(a,b,c)=c \end{cases} \quad (2)$$

수직방향의 b, 좌측상단에서 우측하단으로의 대각선 a, 그리고 우측상단에서 좌측하단으로의 대각선 c의 값을 구하기 위하여 식 (1)과 같이 계산하여 식 (2)와 같이 가장 작은 차이 값을 가지는 방향으로의 두 화소를 평균하여 현재의 화소를 보간한다. 여기서 i, j는 보간하려는 현재화소의 위치를 나타내며, X(i,j)는 현재의 화소 값을 나타낸다. 이 방법은 기본적으로 에지를 3가지 방향으로만 고려하기 때문에 정확한 에지를 찾기는 쉽다나 연산이 간단하여 많이 쓰이는 방법이다.

2.2 E-ELA (Enhanced ELA) 알고리즘

E-ELA 알고리즘은 ELA 알고리즘과 거의 비슷하지만 p, q를 이용하여 좀 더 정확한 에지 방향을 예측 할 수 있다는 장점을 가지고 있다.

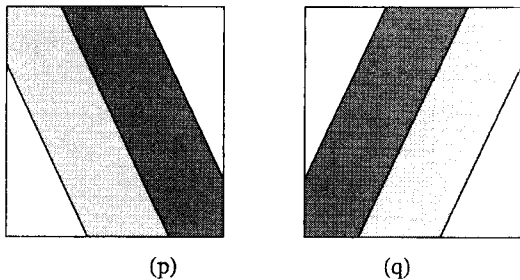


그림 2. p와 q의 대각선 방향 패턴

$$\begin{cases} p = |X(i-1,j-1) - X(i+1,j)| + |X(i-1,j) - X(i+1,j+1)| \\ q = |X(i-1,j+1) - X(i+1,j)| + |X(i-1,j) - X(i+1,j-1)| \end{cases} \quad (3)$$

여기서 p는 좌측상단에서 우측하단으로, q는 우측상단에서 좌측하단으로의 에지를 나타낸다. 즉, p가 나타내는 에지는 ELA 알고리즘에서의 a와, q는 c와 의미상 같다. 하지만 구하는 과정이 ELA 알고리즘 과정과는 달리 좌측상단에서 하단간의 화소차이 값과 상단에서 우측하단의 화소차이 값을 합함으로써 p의 값을 구한다. 이와 비슷하게 q도 우측상단에서 하단간의 화소차이 값과 상단에서 좌측하단의 화소차이 값을 합하여 구한다. E-ELA 알고리즘은 아래의 식(4)와 같이 정의된다.

$$X(i,j) = \begin{cases} (X(i-1,j-1)+X(i+1,j+1))/2, & \text{if } p < q \text{ and } \min(a,b)=a \\ & \text{if } p = q \text{ and } \min(a,b,c)=a \\ (X(i-1,j)+X(i+1,j))/2, & \text{if } p < q \text{ and } \min(a,b)=b \\ & \text{if } p = q \text{ and } \min(a,b,c)=b \\ (X(i-1,j+1)+X(i+1,j-1))/2, & \text{if } p > q \text{ and } \min(b,c)=b \\ & \text{if } p = q \text{ and } \min(a,b,c)=c \end{cases} \quad (4)$$

위의 식은 ELA 보다는 조건이 좀 더 복잡하다는 것을 알 수 있다. 기본적으로 ELA와 같이 3가지 에지를 고려하지만 에지 예측을 더욱 세밀히 고려하고 있다. 우선 p, q의 대소비교를 통하여 p가 q보다 작은 경우는, a와 b의 대소를 비교한다. 만약 a의 값이 b보다 작을 경우는, 대각선 a의 방향으로 현재의 화소를 보간하지만, b가 더 클 경우는 b의 방향으로 현재와 화소를 보간한다. 반대로 p와 q의 값 중에서 p가 더 클 경우는 b와 c의 대소를 비교하여 위와 비슷한 방법으로 현재와 화소를 보간한다. 마지막으로 p와 q의 값이 같은 경우는, a, b 그리고 c 중에서 가장 작은 값을 가지는 방향으로 보간하는 방법이다. E-ELA방법은 ELA와 처리방법이 거의 흡사하나 에지를 더욱더 정확히 예측함으로써 표 3에서 볼 수 있듯이 객관적인 PSNR이 더욱 좋을 수 있을 뿐만 아니라, 그림 9와 같이 주관적인 화질도 향상되었음을 알 수 있다.

2.3 DOI (Direction-Oriented Interpolation) 알고리즘

ELA나 E-ELA 알고리즘은 보간하려는 화소의 바로 상단라인과 하단라인을 참조 하여 현재의 화소를 보간하지만, DOI 알고리즘은 상단 및 하단 각각 2라인씩을 참조하여 현재의 화소를 보간하는 방법을 사용한다.

DOI 알고리즘은 기본적으로 3x3블록 단위로 계산을 하여 SDV_U와 SDV_L를 구하는 방법을 사용한

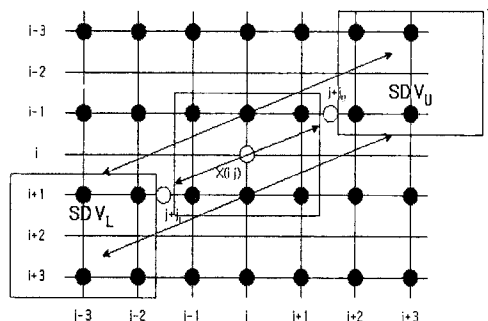


그림 3. DOI 알고리즘 실행

다. 여기서, SDV_U 와 SDV_L 가 각각 나타내는 의미는 보간하려는 현재의 화소의 에지가 상단으로는 SDV_U 방향으로 하단으로는 SDV_L 방향으로 예측한 것을 나타낸다. 즉, 에지의 상단 부분과 하단 부분을 따로 구한다. 상단벡터 SDV_U 와 하단벡터 SDV_L 는 아래 식 (5)와 (6)을 통하여 얻어진다.

$$S_U(k) = \sum_{m=-1}^1 (|X(i-1, j+m) - X(i-3, j+m+k)|^2 + |X(i+1, j+m) - X(i-1, j+m+k)|^2) \quad (5)$$

$$S_L(k) = \sum_{m=-1}^1 (|X(i+1, j+m) - X(i+3, j+m+k)|^2 + |X(i-1, j+m) - X(i+1, j+m+k)|^2)$$

$$\begin{cases} SDV_U = \arg \min(S_U(k)), & -R \leq k \leq R \\ SDV_L = \arg \min(S_L(k)), & -R \leq k \leq R, \text{ where } R=16 \end{cases} \quad (6)$$

여기서 R 은 에지를 예측하기 위한 탐색범위로서 적절히 16을 채택했다. k 는 탐색범위 안의 변수로써 상단벡터 SDV_U 와 하단벡터 SDV_L 을 -16부터 +16 범위 안에서 각각 찾는데 이용된다. 식 (5)는 3x3블록 단위로 보간하려는 현재의 블록과 상단 및 하단 블록과의 각각의 화소의 차이값을 제공하여 합하는 SSE(Sum of Square Error)를 사용한다. 즉, S_U (Sum of Block Difference in Upper Line)들을 구하기 위하여 현재 블록과 상단 블록과의 SSE값을 각각 $S_U(-16)$ 부터 $S_U(16)$ 까지 할당하게 된다. 이와 마찬가지로 S_L (Sum of Block Difference in Lower Line)도 현재 블록과 하단블록과의 SSE값을 각각 $S_L(-16)$ 부터 $S_L(16)$ 까지 할당하게 된다. 식 (6)은 33개의 S_U 값들 중에서 최소의 값을 가지는 방향 값을 SDV_U 에 할당한다. 마찬가지로 33개의 S_L 값들 중에서 최소의 값을 가지는 방향 값을 SDV_L 에 할당한다. 현재 보간하려는 화소의 에지가 상단으로는 SDV_U 방향으로 하단으로는 SDV_L 방향으로 예측되었다는 것을 의미한다.

그림 4는 DOI 알고리즘 순서도를 나타낸다. 우선 보간하려는 화소의 바로 위아래 화소의 차이 값이 임계값인 T(10)보다 작은 경우는 바로 두 개의 화소의 평균값으로 현재의 화소를 보간하고 큰 경우는 식 (5), (6)에 의해 구한 SDV_U , SDV_L 을 이용하여 현재의 화소를 구한다. 일반적으로 SDV_U , SDV_L 는 각각 절대값은 같고 부호는 반대인 경우가 에지를 정확히 찾은 것으로 간주된다. 하지만 1의 오차는 허용하여 $|SDV_U + SDV_L| < 1$ 경우는 에지 예측이 거의 정확하게 찾은 것으로 간주한다. 이 경우 상단벡터와 하단벡터를 각각 2로 나누어 바로 위아래 라인을 참조하는데, 2로 나누는 이유는 그림 3

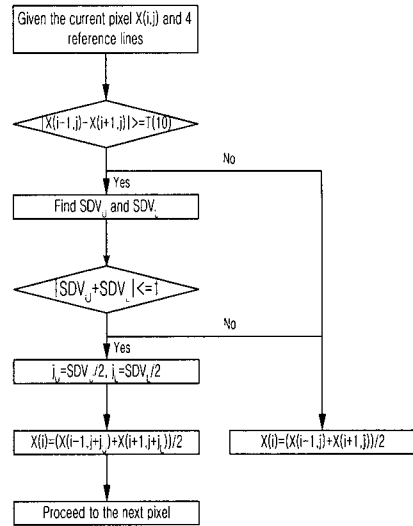


그림 4. DOI 알고리즘 순서도

을 보면 쉽게 이해 될 수 있다. 이 경우 상단벡터와 하단벡터가 짝수 값이 나오면 2로 나누어도 정수가 되기 때문에 그에 해당하는 화소 값을 사용하면 되지만 홀수 값인 경우는 2로 나누면 정수가 아니기 때문에 양옆에 있는 값을 평균하여 사용한다. 예를 들면, 상단벡터가 6이고 하단벡터가 -5인 경우 상단벡터를 2로 나누면 3이되어 보간하려는 위의 라인의 오른쪽 3번째 화소 값을 사용하지만 하단벡터는 2로 나누면 -2.5가 되어 아래라인의 왼쪽 3번째와 2번째 화소 값을 평균하여 -2.5위치의 화소 값을 만들어낸다. 마지막으로, 위의 라인 3의 위치의 화소 값과 아래라인 -2.5의 위치의 화소 값을 평균하여 현재의 화소 값으로 보간한다. $|SDV_U + SDV_L| < 1$ 가 아닌 경우는 에지 예측이 잘못된 것으로 간주하여 보간하려는 현재의 화소가 공간적 상관성이 가장 높은 위아래 화소의 평균값으로 보간한다. DOI 알고리즘은 에지 예측을 ELA나 E-ELA보다는 정확하게 찾을 수 있어 영상의 특성을 더욱 잘 고려한다. 하지만 그림 5나 표 1에서 볼 수 있듯이 저각도($0 \sim 45^\circ$)의 에지는 상당히 잘 고려하나 고각도

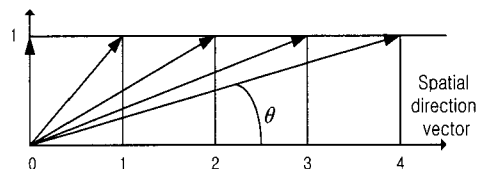


그림 5. DOI 알고리즘 방향각도

표 1. SDV와 방향각도의 상관관계

SDV	0	±1	±2	±3	...	±SDV
Degree(θ)	90°	±45°	±27°	±18°	...	±tan ⁻¹ (1/SDV)

(45° ~ 90°)는 고려를 하지 않는 단점을 가지고 있다. 그래서 복잡하거나 고각도 에지를 많이 포함하는 영상의 경우는 객관적으로나 주관적으로 좋지 않은 결과를 초래 할 수 있다.

그림 5와 표 1은 에지의 각도와 SDV와의 관계를 보여준다.

III. 제안하는 알고리즘

위에서 언급했듯이 De-interlacing 기법은 크게 두 단계의 과정을 거치게 된다. 우선은 에지의 방향성을 결정하게 된다. 다음으로 결정된 에지의 방향을 따라서 잃어버린 화소 값을 보간하게 된다. DOI 알고리즘은 고각도 에지를 찾을 수 없는 단점을 가지고 있었다. 이를 해결하기 위해서 고각도 에지를 고려하는 수직방향벡터인 VDV(Vertical Direction Vectors)를 추가하고 마지막으로 보간하려는 화소는 DOI 알고리즘과는 달리 방향벡터로 6개의 후보 화소를 찾아 중간값(median) 필터를 사용하여 보간한다. 그림 6을 보면, 수평방향벡터 HDV(Horizontal Direction Vectors)을 이용하여 DOI 알고리즘에서 SDV와 같은 기능을 하는 수직(90°)에지와 저각도 에지 방향을 예측 할 수 있다. 그리고 본 논문에서 제안하는 새로운 VDV를 이용하여 고각도 에지 방향을 예측 할 수 있게 된다.

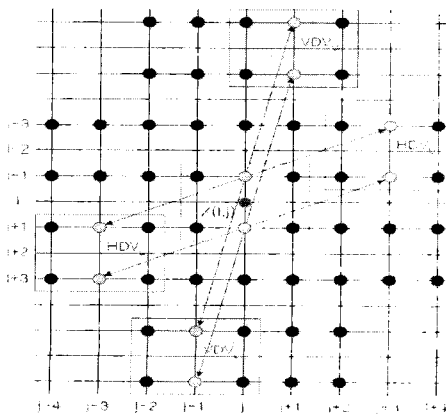


그림 6. 제안하는 알고리즘 실례

그림 6은 제안하는 알고리즘의 실례로써 DOI와 비슷하지만 HDV를 구하기 위하여 수평방향 상하위 2라인씩(i-3, i-1, i+1, i+3) 참조할 뿐만 아니라 VDV를 구하기 위하여 추가로 수직방향 5라인(j-2, j-1, j, j+1, j+2)을 더 참조하여 현재의 화소를 보간 하는데 사용한다. HDV와 VDV를 구하는 식은 아래의 (7), (8)과 같다.

$$\begin{cases}
 SH_U(k) = \frac{1}{m-1} \sum_{m=1} \{ \chi(i-1, j-m) - \chi(i-3, j-m+k) \}^2 + \{ \chi(i-1, j-m) - \chi(i-1, j+m-k) \}^2 \\
 SH_L(k) = \frac{1}{m-1} \sum_{m=1} \{ \chi(i+1, j-m) - \chi(i+3, j-m-k) \}^2 + \{ \chi(i-1, j-m) - \chi(i+1, j+m-k) \}^2 \\
 SV_U(p) = \frac{1}{m-1} \sum_{m=1} \{ \chi(i-1, j-m) - \chi(i-1+p, j-1+m) \}^2 + \{ \chi(i-1, j-m) - \chi(i+1+p, j-1+m) \}^2 \\
 SV_L(p) = \frac{1}{m-1} \sum_{m=1} \{ \chi(i-1, j-m) - \chi(i-1+p, j-1-m) \}^2 + \{ \chi(i-1, j-m) - \chi(i+1+p, j-1-m) \}^2
 \end{cases} \quad (7)$$

$$\begin{cases}
 HDV_U = \arg \min \{ SH_U(k) \}, & -R \leq k \leq R \\
 HDV_L = \arg \min \{ SH_L(k) \}, & -R \leq k \leq R, \text{ k: natural number} \\
 VDV_U = \arg \min \{ SV_U(p) \}, & -R \leq p \leq R \\
 VDV_L = \arg \min \{ SV_L(p) \}, & -R \leq p \leq R, \text{ p: odd number, where } R=16
 \end{cases} \quad (8)$$

note : except when p is ±1 because it is the same case when k is ±1

식 (7)은 식 (5)와 비슷하지만 수평방향벡터(HDV)와 수직방향벡터(VDV)를 구분하기 위하여 식 (5)에서 사용한 공간방향벡터(SDV)를 구하기 위한 S_U, S_L 대신에 SH_U, SH_L로 사용하고, 추가로 수직방향벡터를 구하기 위하여 SV_U, SV_L을 사용한다. 여기서 HDV는 식 (5)에서 SDV와 같다. 단지 2가지의 벡터를 구분하기 위하여 다르게 표현했다. 처리과정은 식 (5)와 유사하지만, VDV를 구하기 위한 변수로 p를 추가로 사용하였다. 탐색범위(Search Range)는 DOI 알고리즘과 마찬가지로 수평방향으로 16, 수직방향으로도 16을 사용하였다. 하지만, 수평방향의 경우는 -16부터 16까지 33가지의 경우를 고려 할 수 있으나 수직방향으로는 격행으로 주사되어 있는 특성상 짝수를 고려 할 수는 없다. 그리고 SV가 1인 경우는 SH가 1인 경우와 겹치게 되어 제외하였다. 즉, 표 2에서 볼 수 있듯이 수평방향으로는 -16, -15, ..., -1, 0, 1, ..., 15, 16와 같이 에지를 33가지 경우를 고려하나 수직방향으로는 -15, -13, ..., -5, -3, 3, 5, ..., 13, 15와 같이 에지를 14가지 경우를 고려 할 수 있다. 식 (8)은 식 (7)에서 각각 구한 HDV_U, HDV_L 각각 33가지 경우와 VDV_U, VDV_L 각각 14가지 경우 중에서 가장 작은 값을 가지는 방향을 HDV_U, HDV_L, VDV_U, VDV_L 각각 할당한다.

$$\begin{cases} a=X(i-3,j+HDV_U) \\ b=X(i+1,j+HDV_L) \\ c=X(i-1,j+HDV_U) \\ d=X(i+3,j+HDV_L) \end{cases} \quad (9)$$

$$\begin{cases} e=X(i-1+VDV_U,j+1) \\ f=X(i-1+VDV_U,j-1) \\ g=X(i+1+VDV_U,j+1) \\ h=X(i+1+VDV_U,j-1) \end{cases} \quad (10)$$

식 (9)에서 a, c는 HDV_U에 의하여 결정되는 화소 값이고 b, d는 HDV_L에 의하여 결정되는 화소 값이다. 유사하게, 식 (10)에서 e, g는 VDV_U에 의하여 결정되는 화소 값이고 f, h는 VDV_L에 의하여 결정되는 화소 값이다. 이 값들이 필요한 이유는 DOI알고리즘에서는 에지의 방향을 예측하여 보간 하려는 화소의 바로 위아래 라인을 사용하기 때문에 SDV_U, SDV_L값을 2로 나누어 그에 해당하는 화소의 값의 평균으로 보간하였으나 제안하는 알고리즘에서는 그림 6에서 볼 수 있듯이 방향벡터가 가리키는 4개의 후보 화소와 거리가 가장 짧아서 공간적 상관성이 가장 높은 바로 위아래 화소 2개 총 6개의 후보 화소를 가지고 중간값 필터를 사용하여 보간한다. 자세한 흐름은 그림 7의 제안하는 알고리즘 순서도에서 알 수 있다.

제안하는 알고리즘은 9개의 라인(i-3, i-1, i+1, i+3, j-2, j-1, j, j+1, j+2)을 참조한다. DOI 알고리즘에서 처음에 하는 처리과정인 보간하려는 바로 위아래 화소 값인 X(i-1,j), X(i+1,j) 차이 값을 임계

값 T(10)와 비교한다. 만약 차이 값이 임계값보다 작다면 X(i-1,j), X(i+1,j)의 평균값으로 현재 화소 X(i,j)값을 보간하고 그렇지 않은 경우에는 식 (7), (8)로 HDV_U, HDV_L, VDV_U, VDV_L 4개의 방향벡터를 찾는다. 수평방향벡터 HDV들을 사용할지 수직방향벡터 VDV들을 사용할지는 SH값들과 SV값들을 비교하여 결정한다.

그림 7의 SH_U(HDV_U), SH_L(HDV_L), SV_U(VDV_U), 그리고 SV_L(VDV_L)가 각각 뜻하는 의미는 SSE로 구한 값 SH, SV값들 중에서 최소의 값을 나타낸다. SH_U(HDV_U) < SV_U(VDV_U) 그리고 SH_L(HDV_L) < SV_L(VDV_L) 인 경우는 에지가 수직방향벡터 보다는 수평방향벡터가 상관성이 높다는 것을 뜻한다. 따라서, 이 경우는 수평방향벡터 HDV값들을 사용한다. DOI 알고리즘에서와 마찬가지로 에지를 정확히 예측했다면 HDV_U, HDV_L은 각각 절대 값은 같고 부호는 반대이다. 하지만 1의 허용치를 두어서 |HDV_U+HDV_L|<=1 인 경우 식 (9)로 구한 화소 값 a, b, c 그리고 d개와 현재의 화소와 공간적 상관성이 가장 높은 X(i-1,j), X(i+1,j) 2개 총 6개 후보화소 값을 중간값 필터를 사용하여 현재화소를 보간한다. 여기서 중간값 필터처리 과정이라 함은 홀수개의 값이 있는 경우에는 크기별로 정렬하여 중간에 있는 값을 채택하여 사용하면 되지만 여기에서는 짝수개의 후보 화소 값이 존재하여 크기별로 정렬하여 중간에 있는 2개를 평균하여 현재의 화소를 보간하는 방법을 사용하였다.

SV_U(VDV_U) < SH_U(HDV_U) 그리고 SV_L(VDV_L) < SH_L(HDV_L) 인 경우는 위와 반대로 수평방향벡터 보다는 수직방향벡터가 상관성이 높다는 것을 의미한다. 이 경우도 위와 비슷하게 처리를 하고 마지막으로 중간값 필터를 식 (10)로 구한 화소 값 e,f,g 그리고 h 4개와 X(i-1,j), X(i+1,j) 2개 총 6개 후보화소 값을 중간값 필터를 사용하여 현재 화소를 보간한다.

위 2가지도 아닌 경우 SH_U(HDV_U) < SV_U(VDV_U) 그리고 SH_L(HDV_L) > SV_L(VDV_L) 와 같은 에지예측을 잘못 한 것으로 간주하여 공간적 상관성이 가장 높은 X(i-1,j), X(i+1,j) 값의 평균으로 현재 화소를 보간한다.

그림 8과 표 2는 에지의 각도와 방향벡터와의 관계를 나타낸다. 기존 DOI 알고리즘은 방향벡터 SDV를 이용하여 수직방향(90°)과 저각도 에지(0° ~ 45°)를 고려하나 고각도 에지(45° ~ 90°)는 고려하지 못하는 단점이 있었다. 하지만 제안하는 알고리즘은 VDV들을 추가하여 고각도 에지까지 고려를 했다.

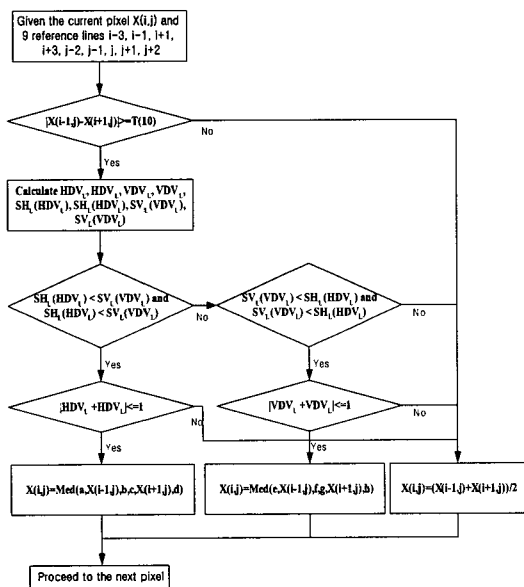


그림 7. 제안된 알고리즘 순서도

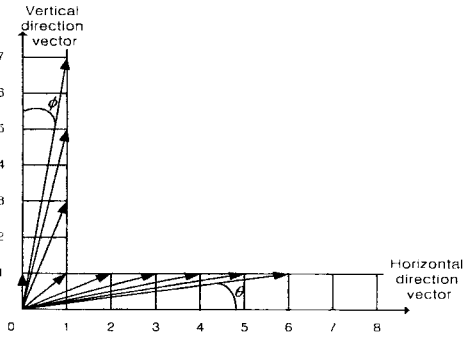


그림 8. 제안된 알고리즘 방향벡터

표 2. HDV, VDV와 방향각도의 상관관계

HDV	0	±1	±2	±3	...	±HDV
Degree(°)	90°	±45°	±27°	±18°	...	$\pm \tan^{-1}(1/HDV)$
VDV	±3	±5	±7	±9	...	±VDV
Degree(°)	±72°	±79°	±82°	±84°	...	$\pm \tan^{-1}(VDV)$

Note: ±1 of VDV do not need because it is the same case when HDV is ±1

IV. 실험 결과 및 분석

이번 장에서는 제안하는 알고리즘을 객관적인 성능과 주관적인 화질 측면에서 기존의 알고리즘과 비교하게 된다. 실험은 Barbara 영상을 포함해서 7개의 정지 영상을 대상으로 하였다. 512×512 크기의 원본 영상을 짝수 라인만을 복사하고 홀수라인은 원래 영상에서 없앤 512×256로 down-sampling된 격행 주사를 만든 후 기존 및 제안하는 알고리즘으로 홀수 라인을 보간하기 위해서 적용한다.

표 3은 다양한 정지 영상에서 기존의 알고리즘과 제안하는 알고리즘의 객관적 결과(PSNR)들을 보여

표 3. 다양한 De-interlacing 기법을 통한 각각의 영상의 PSNR 비교

Image	Method				
	LD	ELA	E-ELA	DOI	Proposed
Baboon	21.62	22.99	23.18	23.43	23.51
Barbara	27.24	25.20	30.62	29.58	32.05
Airplane	28.60	31.15	31.30	31.71	31.83
Goldhill	30.61	32.18	32.49	33.45	33.58
Zelda	36.71	34.05	34.18	33.93	34.19
Boats	35.56	39.22	39.77	41.36	41.66
Finger	24.68	28.94	29.41	29.29	31.63

준다. 위의 표에서 알 수 있듯이 제안하는 알고리즘이 기존의 알고리즘들 보다 객관적인 성능이 뛰어난 것을 알 수 있다. 특히, 다른 영상보다 더 복잡하거나, 고각도 에지(45° ~ 90°)를 많이 포함하고 있는 Barbara 영상의 경우 DOI 알고리즘과 비교하면 PSNR이 2.47dB까지, Finger 영상의 경우는 2.34dB까지 향상되었음을 알 수 있다. 즉, 제안된 알고리즘을 단순한 영상보다는 복잡한 영상에 적용할수록

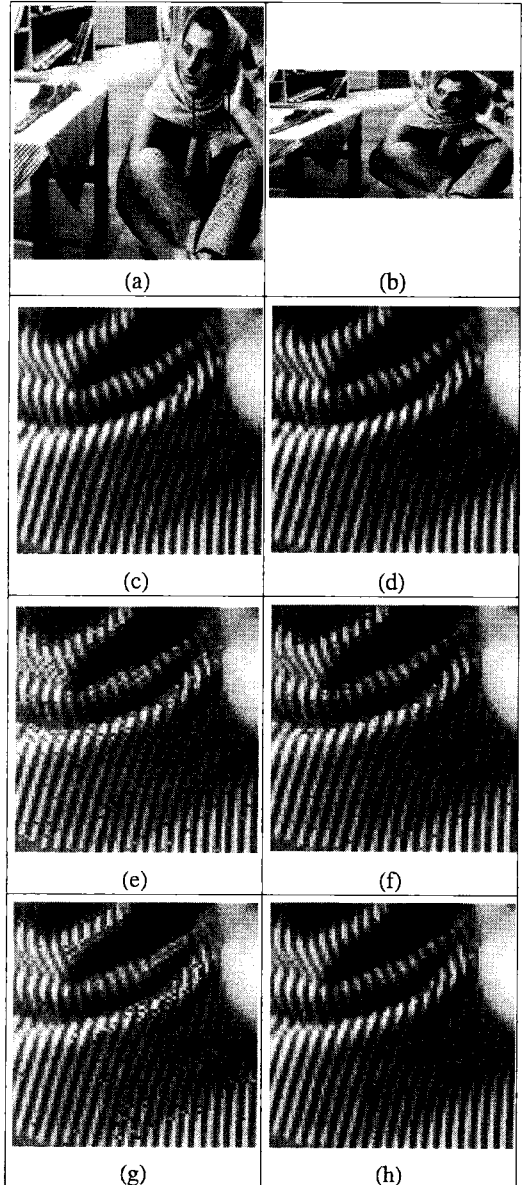


그림 9. (a)Barbara 원본 영상, (b) Barbara down-sampling된 영상 (c)원본 선택부분 확대영상, (d)LD 적용 확대 영상, (e)ELA 적용 확대 영상, (f)E-ELA 적용 확대 영상, (g)DOI 적용 확대 영상, (h)제안된 알고리즘 적용 확대 영상

더욱 좋은 성능을 낼 수 있음을 보여 준다. 다음은 주관적인 측면에서 살펴보도록 하겠다.

그림 9는 다양한 알고리즘을 적용시켜 얻은 영상들을 특정부분 확대한 것이다. (a)의 영상은 원본 영상에 확대할 부분을 사각형으로 표시를 해 놓았다.

(b)는 원본 영상을 down-sampling한 영상으로써 잃어버린 라인을 보간 하기 위하여 De-interlacing 알고리즘을 적용시키는 것이다. 위 영상의 확대한 부분은 고각도 에지를 많이 포함하고 있어 제안된 알고리즘의 특성을 나타내는데 효과적이다. LD(Line Double) 알고리즘을 적용시킨 영상의 경우 원본영상과 많이 비슷해 보이지만 PSNR의 경우는 상당히 낮다. 이는 LD알고리즘의 경우 에지의 특성을 파악하지 않고 라인을 복사하여 없앤 라인을 채우는 방법으로써 ELA나 E-ELA처럼 뚜렷하게 보이는 에러는 없어 보이지만 객관적 성능이 상당히 낮아진다. 즉, 주관적 화질은 인간의 눈으로 판단하는 것이므로 객관적 성능이 높다하여 항상 주관적 화질까지 좋은 것은 아니고 주관적 화질이 좋다하여 객관적 성능까지 높은 것은 아니다. 하지만 제안된 알고리즘은 그림 (9)의 (h)에서 볼수 있듯이 객관적 성능이 높을 뿐만 아니라 에러가 거의 보이지 않는 주관적 화질까지 동시에 좋은 것을 알 수 있다.

V. 결 론

많은 Intra-field De-interlacing 방법에서 중요한 부분은 정확히 에지의 방향을 찾아서 효율적으로 보간하는 것이다. 기존의 알고리즘은 고각도 에지를 고려하지 않아서 화질열화를 초래하였지만 본 논문에서는 $0 \sim 45^\circ$ 사이의 저각도 뿐만 아니라 VDV를 추가하여 $45 \sim 90^\circ$ 사이의 고각도 에지를 고려하는 전 방향 에지를 예측한다. 그리고 마지막 보간하는 과정에서 DOI 알고리즘은 보간하려는 화소의 위아래 라인에서 에지를 예측한 방향에 해당하는 두 화소의 평균값으로 보간을 한다. 반면, 제안하는 알고리즘은 마지막 보간하는 과정에서 위아래 2라인만을 사용하지 않고 HDV 또는 VDV에 해당하는 4개의 후보화소 값과 보간하려는 화소와 위아래 화소 2개 총 6개의 화소를 정렬하여 중간의 2개의 값을 평균내어 보간하는 중간값 필터를 사용한다. 중간값 필터는 에러를 상당히 줄이는 특성을 가지고 있어서 보간할 때 이를 사용하여 에러를 줄여서 화질열화를 막을 수 있다. 표 3과 그림 9에서 알 수 있듯이 제안하는 알고리즘이 객관적인 성능(PSNR) 뿐만 아니라 주관적인 영상 화

질까지 우수하다는 것을 보여준다. 이를 통해 기존의 Intra-field De-interlacing 방식에서 문제가 되었던 에지 성분을 비롯하여 영상의 정보가 손실되는 문제를 개선하였다.

참 고 문 헌

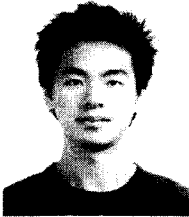
- [1] G. de Haan and E. B. Bellers, Deinterlacing - an overview, Proceedings of the IEEE, 86(9), 1998, 1839-1857.
- [2] T. Doyle, Interlaced to sequential conversion for EDTV applications, Proc. 2nd Int. Workshop Signal Processing of HDTV, L'Aquila, IT, 1998, 412-430.
- [3] M.-H. Lee, J.-H. Kim, K.-K. Ryo, and D.-I. Song, A new algorithm for interlaced to progressive scan conversion based on directional correlation and its IC design, IEEE Trans. Consumer Electronics, 40(2), 1994, 119-125.
- [4] C. J. Kuo, C. Liao, and C. C. Lin, Adaptive interpolation technique for scanning rate conversion, IEEE Trans. Circuits Systems for Video Technology, 6(3), 1996, 317-322.
- [5] T. Chen, H. R. Wu, and Z. H. Yu, An efficient edge line average interpolation algorithm for de-interlacing, Proc. Visual Communications and Image Processing, SPIE 4067, Perth, AU, 2000, 1551-1558.
- [6] Tao Chen, Hong Ren Wu and Zheng Hua Yu, Efficient de-interlacing algorithm using edge-based line average interpolation, Optical Engineering, 39(8), 2000, 2101-2105.
- [7] Hoon Yoo and Jechang Jeong, Direction-oriented interpolation and its application to de-interlacing, IEEE Transactions on Consumer Electronics, 48(4), 2002, 954-962.
- [8] Hyo-Sub Oh, Yoon Kim, You - Young Jung, Aldo W.Morales and Sung-Jea Ko, Spatio-temporal edge-based median filtering for deinterlacing, IEEE International Conference on Consumer Electronics, 2000, 52-53.
- [9] Mei-Juan Chen, Chin-Hui Huang and Ching-Ting Hsu, Efficient De-interlacing Technique by inter-filed information, IEEE Transaction on Consumer Electronics, 50(4),

2004, 1202-1208.

- [10] Kenji Sugiyama, Yoshiyuki Yamada, Naoya Sagara, Improvement of Motion Compensated Inter-Field Interpolation Method for De-Interlacing, TENCON 2006. 2006 IEEE Region 10 Conference, HK, 2006, 1-4.
- [11] Byeong-Doo Choi, Jong-Woo Han, Chan-Su Kim, Sung-Jea Ko, Motion-Compensated Frame Interpolation Using Bilateral Motion Estimation and Adaptive Overlapped Block Motion Compensation, IEEE Trans. Circuits and Systems for Video Technology, 17(4), 2007, 407-416.

구 수 일 (Suil Ku)

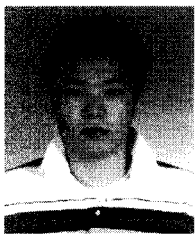
준회원



2007년 2월 한양대학교 전자전기
컴퓨터공학부 졸업
2007년 3월~현재 한양대학교 전
자통신컴퓨터공학과 석사과정
<관심분야> 화질개선, 영상압축

이 세 영 (Se-young Lee)

준회원



2007년 2월 한양대학교 전자전기
컴퓨터공학부 졸업
2007년 9월~현재 한양대학교 전
자통신컴퓨터공학과 석사
<관심분야> 화질개선, 영상압축

강 근 화 (Kunhwa Kang)

준회원



2008년 2월 부산외국어대학교 디
지탈정보공학부 졸업
2008년 3월~현재 한양대학교 전
자컴퓨터통신공학과 석사과정
<관심분야> 영상처리, 영상압축

정 제 창 (Jechang Jeong)

정회원



1980년 2월 서울대학교 전자공학
과 졸업
1982년 2월 KAIST 전기전자 공
학과 석사
1990년 미국 미시간대학 전기 공
학과 공학박사
1980~1986 KBS 기술연구소 연
구원(디지털 TV 및 뉴미디어 연구)
1990년 미국 미시간대학 전기공학과 공학박사
1980년~1986년 KBS 기술연구소 연구원(디지털 TV
및 뉴미디어 연구)
1990년~1991년 미국 미시간대학 전기공학과 연구교수
(영상 및 신호처리 연구)
1991~1995 삼성전자 멀티미디어 연구소 (MPEG,
HDTV, 멀티미디어 연구)
1995~현재 한양대학교 전자통신컴퓨터공학과 교수
(영상통신 및 신호처리 연구실)
1998년 11월 27일 과학기술자상 수상
1998년 12월 31일 정보통신부장관상 표창
<관심분야> 영상처리 및 영상압축