지역 복잡도 기반 방법 선택을 이용한 적응적 디인터레이싱 알고리듬

준회원 홍 성 민*, 정회원 박 상 준**, 정 제 창**°

Adaptive De-interlacing Algorithm using Method Selection based on Degree of Local Complexity

Sung-Min Hong* *Associate Member*, Sang-Jun Park**, Jechang Jeong**° *Regular Members*

요 약

본 논문에서는 영상의 지역 특성별로 보간 방법을 적응적으로 선택하여 적용하는 효과적인 디인터레이싱 알고리듬을 제안한다. 기존의 알고리듬들의 경우 각기 다른 방법으로 방향성을 구하기 때문에 영상의 지역 특성별로 성능이 다르게 나오는 경우가 있다. 또한, FDD(Fine Directional De-interlacing) 알고리듬의 경우 PSNR(Peak Signal-to-Noise Ratio)은 다른 알고리듬들에 비해 높게 나오지만 계산량이 많다는 단점이 있다. 이를 보안하기 위해 본 논문에서는 여러 영상들에서 계산량은 적으면서 화질 성능은 뛰어난 LA(Line Average), MELA(Modified Edge-based Line Average), LCID(Low-Complexity Interpolation Method for De-interlacing) 알고리듬들 중 지역 복잡도(DoLC, Degree of Local Complexity)별로 효과적인 알고리듬을 학습하여 이를 이용하여 보간을 수행하는 디인터레이싱 방법을 제안한다. 실험 결과 제안하는 방법은 좋은 성능에 비해 계산량이 적은 LCID 알고리듬과 비슷한 계산량을 보이면서 객관적 화질이 우수한 FDD, MELA 알고리듬보다 PSNR로 대표되는 객관적 화질과 주관적 화질 측면에서 우수한 결과를 나타내는 것을 알 수 있다.

Key Words : De-Interlacing, Adaptive, Degree of Local Complexity, Interpolation, Method Selection

ABSTRACT

In this paper, we propose an adaptive de-interlacing algorithm that is based on the degree of local complexity. The conventional intra field de-interlacing algorithms show the different performance according to the ways which find the edge direction. Furthermore, FDD (Fine Directional De-interlacing) algorithm has the better performance than other algorithms but the computational complexity of FDD algorithm is too high. In order to alleviate these problems, the proposed algorithm selects the most efficient de-interacing algorithm among LA (Line Average), MELA (Modified Edge-based Line Average),and LCID (Low-Complexity Interpolation Method for De-interlacing) algorithms which have low complexity and good performance. The proposed algorithm is trained by the DoLC (Degree of Local Complexity) for selection of the algorithms mentioned above. Simulation results show that the proposed algorithm not only has the low complexity but also performs better objective and subjective image quality performances compared with the conventional intra-field methods.

※ 본 연구는 2011년도 BK21 사업에 의하여 지원되었음
* 한양대학교 지능형로봇학과 영상통신 및 신호처리 연구실(sungmin0215@gmail.com)
** 한양대학교 전자통신컴퓨터공학과 영상통신 및 신호처리 연구실(sjp21c@gmail.com, jjeong@ece.hanyang.ac.kr) (°:교신저자)
논문번호 : KICS2010-10-485, 접수일자 : 2010년 10월 12일, 최종논문접수일자 : 2011년 2월 28일

Ⅰ. 서 론

현재 통용되고 있는 아날로그 TV시스템은 전송 대역폭의 제한으로 홀수, 짝수 필드가 1/60초의 간격으로 교차되어 하나의 프레임을 구성하는 격행 주사(interlaced)방식을 사용하고 있다. 하지만 최근 많이 쓰이는 HDTV나 PDP, LCD등의 디스플레이 장치에서 격행 주사 방식을 그대로 사용하면 주사선 사이의 깜박거림(flicker) 현상이나 해상도가 떨어지는 현상 등의 심각한 화질열화가 발생한다. 이러한 문제점들을 극복하기 위해 지금까지 다양한 디인터레이싱 방법들이 제안되고 있다.

디인터레이싱 방법은 크게 현재 필드만을 이용하는 공간적 디인터레이싱 방법(Intra-field de-interlacing)[3]-[12]과 현재 필드뿐만 아니라 이전필드까지 여러 장의 필드를 이용하는 시간적 디인터레이싱 방법(Inter-field de-interlacing)으로 나눌 수 있다[1]-[2]. 시간적 디인터레이싱 방법의 경우 공간적 디인터레이싱 방법과 비교하여 우수한 성능을 보이지만 연산량이 많아 하드웨어 구조가 복잡해지고 정확한 움직임 검출에 실패했을 경우 심각한 화질열화를 가져오는 단점이 있다. 반면에 공간적 디인터레이싱 방법은 하드웨어 구현이 용이하고 현재 필드만을 사용하기 때문에 메모리사용에 있어서 유리하다. 또한 시간적 디인터레이싱 방법의 경우 공간적 디인터레이싱 방법을 기본적으로 사용하기 때문에 디인터레이싱 방법의 성능을 향상시키기 위해서는 효율적인 공간적 디인터레이싱 방법이 꼭 필요하다. 일반적으로 공간적 디인터레이싱 방법 중 가장 많이 쓰이는 방법은 ELA(Edge-based Line Average) 알고리듬[3]이다. ELA 알고리듬은 간단한 연산으로 구현이 용이하기 때문에 많이 사용 되고 있지만 제한된 각도(45,90,135도)만을 사용하기 때문에 이 각도를 벗어나는 방향에 대해서는 제대로 된 에지정보를 찾지 못하는 단점이 있다. 이런 단점을 보완하기 위해서 많은 디인터레이싱 방법들이 제안되어 왔다. 그 중 하나가 EELA(Efficient ELA) 알고리듬[4]이다. EELA 알고리듬은 ELA 알고리듬보다 에지검출을 구체적으로 찾음으로써 ELA 알고리듬의 단점을 보완했지만 이 방법 역시 검출 범위가 제한되어있다. 이러한 단점들을 극복하고자 DOI(Directional-Oriented Interpolation)[5], LCID[7], MELA[8], FDD[9] 등의 많은 알고리듬들이 제안되었다. DOI 알고리듬은 공간적 방향벡터(sdv, spatial direction vector)를 이용하여 현재 보간하는 픽셀 위치에서 바로 위, 아래 라인뿐만 아니라 그 위, 아래 라인까지 고려하여 방향을 판단하고 보간을 수행한다. 고려하는 라인이 늘어난 만큼 성능은 올라가지만 그만큼 계산량이 많아지는 단점이 있다. LCID 알고리듬은 복잡도를 최대한 낮추면서 성능을 올리기 위해서 EELA 알고리듬에서 고려하는 방향 외에 수직 및 수평 방향을 더 고려하여 보간을 수행한다. MELA 알고리듬도 LCID 알고리듬과 비슷하게 EELA 알고리듬에서 세로방향을 더 고려하지만 에지 방향을 최초 판단 후 한 번 더 하기 때문에 좋은 성능을 보이고 있다. FDD 알고리듬은 보간할 화소의 주변 화소들이 갖는 공간적 방향성의 경향을 구하고 구해진 경향에 맞게 소벨 연산을 적응적으로 적용하여 기울기 벡터를 구함으로써 정교한 에지의 방향을 구한다. 이렇게 구해진 정교한 에지 방향에 맞게 보간을 수행하므로 좀 더 선명하고 정확한 영상을 얻을 수 있다. 하지만 FDD 알고리듬 역시 화소 주변 값들에 소벨 마스크를 적용하는 연산량 때문에 복잡도가 높다.

본 논문에서는 위에서 언급한 공간적 디인터레이싱 알고리듬들이 영상의 지역 특성별로 성능이 다르게 나오는 것에 착안하여 계산량이 적고 성능이 좋은 LA, MELA, LCID 방법들을 후보 방법으로 이용하여 공간적 특성마다 성능이 가장 좋은 알고리듬을 적응적으로 선택하여 적용하는 공간적 디인터레이싱 알고리듬을 제안한다. 정확성을 높이기 위해 여러 특성의 이미지들을 트레이닝 시킨다. 이렇게 구해진 정보들로 공간적 특성별 효과적인 알고리듬을 적용하여 보간을 수행하기 때문에 좋은 결과를 얻을 수 있다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 기존의 대표적인 공간적 디인터레이싱 알고리듬을 소개하고 3장에서 제안하는 알고리듬을 설명한다. 4장에서는 여러 가지 정지 영상에 대해 기존 알고리듬들과 비교 한 실험 결과 및 분석을 보여주고 5장에서 결론을 맺는다.

Ⅱ. 기존의 공간적 디인터레이싱 알고리듬

본 장에서는 기존의 공간적 디인터레이싱 방법 중 MELA 알고리듬과 LCID 알고리듬에 대해 간략히 소개한다.

2.1 MELA 알고리듬

그림 1은 MELA와 LCID 알고리듬에서 픽셀을

보간하기 위해 사용되는 6개의 이웃 픽셀들을 보여주고 있다. 그림 1에서 i는 현재 픽셀의 열을 의미한다. MELA 알고리듬은 P, Q, V 세 개의 방향벡터와 세 개의 상관계수(C_{-1}, C_0, C_1,)를 이용하여 현재 픽셀이 위치한 지역의 에지방향을 예측한다.

$$\begin{aligned} P &= (|U(i-1)-L(i)|+|U(i)-L(i+1)|)/2 \\ Q &= (|U(i)-L(i-1)|+|U(i+1)-L(i)|)/2 \\ V &= (|U(i-1)-L(i-1)|+|U(i)-L(i)| \\ &\quad +|U(i+1)-L(i+1)|)/3 \end{aligned} \tag{1}$$

식 (1)은 방향벡터들을 구하기 위한 식을 보여주고 있다. 식에서 U는 Upper 라인, L은 Lower 라인을 뜻한다. MELA 알고리듬은 이 식을 이용하여 방향벡터를 구하고, 방향벡터의 관계에 따라 현재 영역의 에지경향을 정한다. 그 후, 식 (2)에서 보여주고 있는 상관계수들을 이용하여 좀 더 정밀하게 에지방향을 판단한다.

$$\begin{aligned} C_{-1} &= |U(i-1)-L(i-1)| \\ C_0 &= |U(i)-L(i)| \\ C_1 &= |U(i+1)-L(i+1)| \end{aligned} \tag{2}$$

식 (3)은 MELA 알고리듬을 나타내고 있다.

$$x(i)=\begin{cases} \dfrac{U(i-1)+U(i)+L(i)+L(i+1)}{4} \\ \quad if\ \{\min(P,Q,V)=P\}\wedge\{C_{-1}<C_0\} \\ \dfrac{U(i)+U(i+1)+L(i-1)+L(i)}{4} \\ \quad if\ \{\min(P,Q,V)=Q\}\wedge\{C_1<C_0\} \\ \dfrac{U(i)+L(i)}{2},\ otherwise \end{cases} \tag{3}$$

식 (3)에서 $x(i)$는 현재 보간되는 픽셀을 의미한다. 에지가 P나 Q방향으로 검출됐을 경우, 각 방향에 속하는 4개의 픽셀의 평균값으로 보간을 수행한다. 그 외의 경우는 위, 아래의 평균값으로 보간을 수행한다.

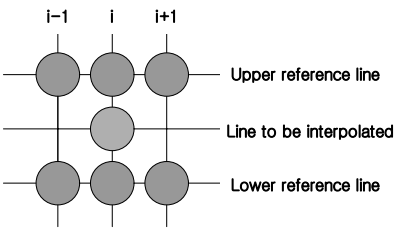


그림 1. 픽셀 보간을 위해서 사용되는 픽셀들

2.2 LCID 알고리듬

LCID 알고리듬은 P, Q 방향의 대각선 방향의 방향벡터 이외에도 수직방향과 수평방향벡터까지 이용하여 픽셀을 보간한다.

그림 2는 LCID 알고리듬에서 고려하는 에지 방향들을 보여주고 있다. EELA 알고리듬에서 고려하는 대각선 성분 외에도 수직 성분과 수평 성분도 고려하여 에지판단의 정확성을 올리고 있다. 식 (4)는 에지 경향별 보간 하는 식을 보여주고 있다.

$$x(i)=\begin{cases} x(i-1), \quad if\ D_h=0 \\ (U(i-1)+U(i)+L(i)+L(i+1))/4 \\ \quad , if\ \min(D_{d1},D_{d2},D_v)=D_{d1} \\ (U(i)+U(i+1)+L(i-1)+L(i))/4 \\ \quad , if\ \min(D_{d1},D_{d2},D_v)=D_{d2} \\ (U(i)+L(i))/2,\ if\ if\ \min(D_{d1},D_{d2},D_v)=D_v \end{cases} \tag{4}$$

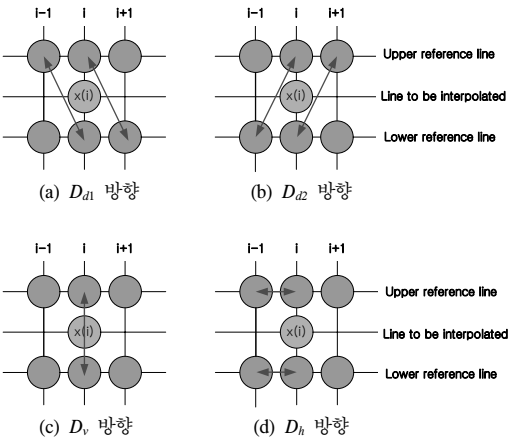


그림 2. LCID 알고리듬에서 사용하는 에지 경향

Ⅲ. 제안하는 알고리듬

서론에서 언급한바와 같이 공간적 디인터레이싱 방법은 잡음에 영향을 잘 받고 에지 판단이 잘못됐을 경우 화질의 열화가 생긴다는 단점이 있지만 간단한 연산량 때문에 실시간 시스템에 많이 사용되고 있다. 이러한 단점을 해결하기 위해서 많은 디인터레이싱 알고리듬들이 제안되었지만 알고리듬별 에

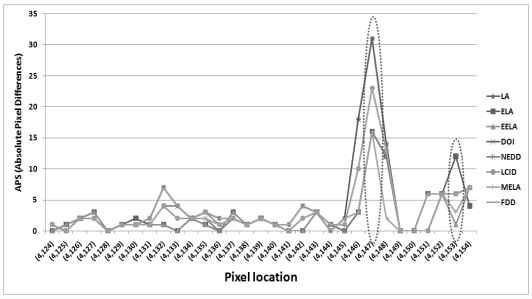


그림 3. "Foreman" 영상의 픽셀 위치별 원본 픽셀과 기존의 디인터레이싱 알고리듬을 이용하여 보간된 픽셀과의 차분치

지를 판단하는 방법이 달라서 영상의 지역 특성별로 성능이 다르게 나오고 있다. 그림 3은 Foreman 영상의 일정 지역에서 기존 알고리듬과 원본 픽셀 값과의 차이를 보여주는 그래프이다. Pixel location은 영상에서 픽셀 좌표를 의미한다. 그래프의 (4, 147) 위치를 보면 FDD, LA 알고리듬이 다른 알고리듬들에 비해 원본과의 차 값이 작지만, (4,153) 위치에서는 EELA 알고리듬의 차 값이 다른 알고리듬들의 차 값보다 작은 것을 볼 수 있다. 이 결과를 통해 알 수 있듯이 영상의 지역 특성별로 알고리듬들의 성능이 다르게 나오고 있다. 이를 이용하여 본 논문에서는 기존의 다양한 공간적 디인터레이싱 알고리듬들 중에서 적은 연산량에 비해 좋은 성능을 내고 있는 MELA와 LCID 알고리듬를 이용한 공간적 디인터레이싱 알고리듬을 제안한다. LA, MELA, LCID 세 알고리듬을 후보 방법으로 이용하여 지역복잡도별로 효과적인 알고리듬을 트레이닝을 통해 찾고 이 정보를 이용하여 3가지 후보 방법들 중에 최적의 방법을 선택하여 픽셀을 보간한다. LA 알고리듬 사용은 MELA와 LCID 알고리듬이 평탄한 영역에서도 수직방향을 찾기 위해 여러 방향을 고려하는 연산량을 줄일 수 있다.

3.1 DoLC

먼저 다양한 이미지들에서 효과적인 알고리듬을 구하기 위해서는 지역별로 방향성의 경향을 구해야 한다. 본 논문에서는 이러한 경향을 구하기 위해 DoLC를 사용하고 있다. 식 (5)은 DoLC를 구하는 식을 보여준다. 식에서 U는 Upper 라인, L은 Lower 라인을 뜻한다.

$$\text{DoLC} = |U(i-1) - L(i-1)| + |U(i) - L(i)| + |U(i+1) - L(i+1)| \quad (5)$$

본 논문에서는 DoLC의 연산량을 줄이기 위해

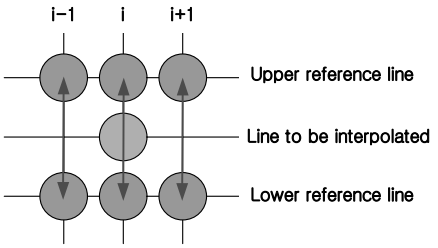


그림 4. 지역 복잡도를 구하기 위하여 사용되는 여섯 개의 픽셀들

그림 4에서 볼 수 있듯이 3x3 윈도우를 사용하여 위, 아래 픽셀들의 차의 합으로 구하고 있다.

3.2 DoLC별 효과적인 알고리듬 선택

이미지에서 보간할 픽셀위치별로 DoLC를 구하고 나면, 그 DoLC의 가장 효율적인 알고리듬을 선택하게 된다. 먼저 현재 보간하는 픽셀을 LA, MELA, LCID 알고리듬을 이용하여 예측하고 식 (6)과 같이 예측된 값과 본래의 픽셀 값과의 차분치를 구하게 된다. 그 후에 그 값을 그 픽셀 위치에서 구한 DoLC에 저장시킨다. 정확성을 올리기 위해서 이 과정을 다양한 특성의 이미지들에서 반복하고 그 차분치를 누적시킨다. $\hat{x}_{LA}$는 LA 알고리듬으로 예측된 값이고 $\hat{x}_{MELA}$는 MELA 알고리듬으로 예측된 값, $\hat{x}_{LCID}$는 LCID 알고리듬으로 예측된 값이다.

$$\begin{aligned} E_{LA} &= \left| x(i) - \hat{x}_{LA} \right| \\ E_{MELA} &= \left| x(i) - \hat{x}_{MELA} \right| \\ E_{LCID} &= \left| x(i) - \hat{x}_{LCID} \right| \end{aligned} \quad (6)$$

세 개의 알고리듬에서 각각 누적된 값들은 DoLC의 값별로 비교를 하게 되는데, 누적된 값의 평균값 $E_{LA,avg.}$, $E_{MELA,avg.}$, $E_{LCID,avg.}$ 중에 가장 작은 값을 가지는 알고리듬이 해당하는 DoLC에서 가장 효율적인 알고리듬으로 선택되게 된다. 선택된 알고리듬은 LUT(LookUpTable)에 저장되고, 만들어진 LUT를 이용하여 보간할 픽셀위치에서 DoLC를 구한 후, 그 값에서 선택된 알고리듬을 이용하여 디인터레이싱을 수행하게 된다. 그림 5는 위에서 설명한 DoLC별 가장 효율적인 방법을 찾기 위한 트레이닝 과정에 대한 블록다이어그램을 보여주고 있다. 차분치를 누적하는 과정에서, 만약 차분치가 차분치 임

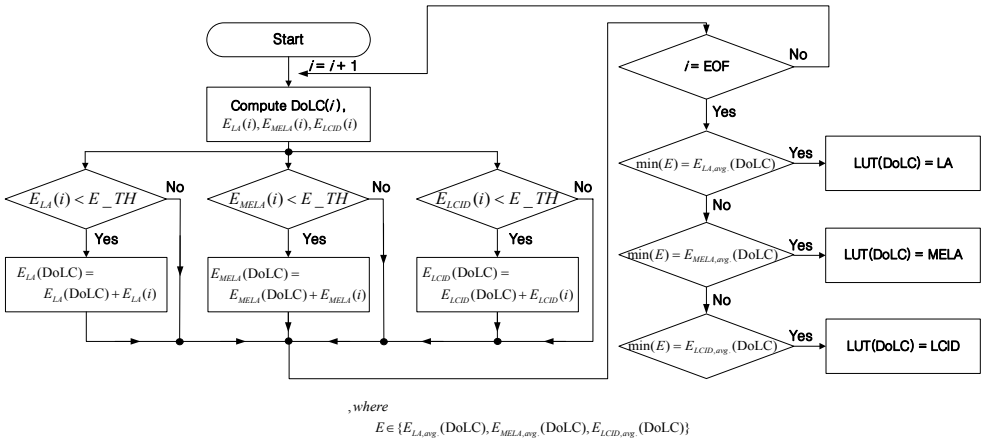


그림 5. 제안하는 알고리듬의 트레이닝 과정에 대한 블록다이어그램

계값(*E_TH*)보다 크거나 같을 경우, 그 차분치의 정확도가 떨어진다고 판단하여 값을 누적시키지 않는다. 차분치가 임계값보다 작을 경우에만 누적시켜, 트레이닝의 정확도를 올렸다. 본 논문에서는 *E_TH*를 79로 세팅하였고, 이 값은 여러 테스트 영상에서 실험을 통하여 결정되었다. 그림 6은 여러 특성을 가지는 20개의 정지영상을 대상으로 DoLC별 상관도가 가장 높은 알고리듬을 보여주는 그래프이다. 0으로 선택된 부분은 LA, MELA, LCID 세 알고리듬의 상관도가 같은 지역으로, 이럴 경우 LA 알고리듬을 선택하게 된다. LA 알고리듬은 에지방향의 고려 없이 보간을 하기 때문에 연산량이 작다. 에지가 없는 평탄한 영역에서는 대부분 수직방향이 선택되기 때문에 이런 부분에서는 LA, MELA, LCID 알고리듬으로 보간한 픽셀과 본래 픽셀과의 차분치가 같은 경우가 많다. 이럴 경우 LA 알고리듬을 쓰게 되면 에지방향을 찾기 위한 연산을 하지 않기 때문에 DoLC를 구하는 연산을 보상할 수 있다.

3.3 제안하는 알고리듬

처음, 보간하는 픽셀 위치에서 DoLC를 구한다. 그 DoLC 값이 임계값(TH)보다 작거나 같을 경우, 현재 픽셀이 평탄한 영역에 있다고 판단되어 그 픽셀의 위, 아래 픽셀들의 평균 값을 이용하여 보간하게 된다. DoLC 값이 TH보다 클 경우에만 DoLC에 해당하는 LUT의 저장된 값을 이용하여 LA, MELA, LCID 알고리듬을 적응적으로 적용시켜 픽셀을 보간한다. 그림 7은 제안하는 알고리듬의 블록다이어그램을 보여주고 있다.

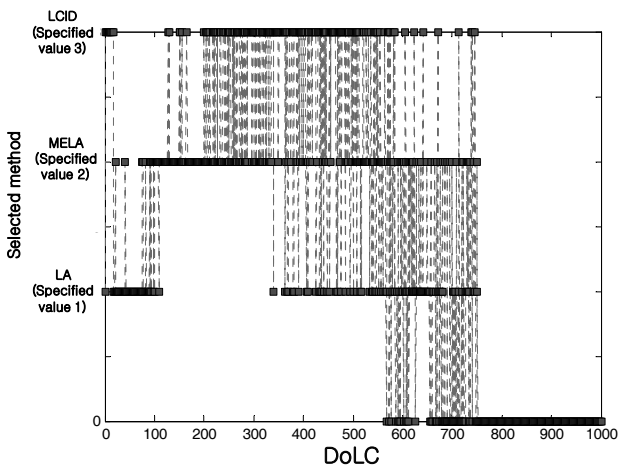


그림 6. DoLC별 선택된 알고리듬

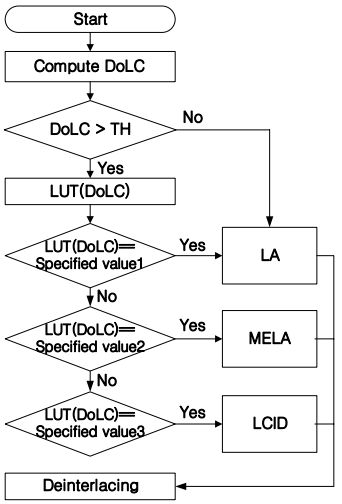


그림 7. 제안하는 알고리듬의 블록 다이어그램

Ⅳ. 실험 결과 및 분석

다양한 크기의 여러 가지 표준 정지 영상을 대상으로 제안하는 알고리듬의 성능을 평가했다. 본 실험에서는 DoLC별 효과적인 알고리듬을 찾기 위해 여러 특성의 다양한 이미지들을 트레이닝 했다. 또한, 기존의 에지 기반의 공간적 디인터레이싱 알고리듬과의 성능을 비교하기 위해서 객관적인 성능지표로 널리 쓰이는 PSNR을 사용하였다. 식 (7)는 PSNR을 구하는 식을 보여주고 있다.

$$\mathrm{PSNR} = 10 \times \log\left(\frac{255^2}{MSE^2}\right) \qquad (7)$$

식 (7)에서 *MSE*는 Mean Squared Error를 나타낸다. 실험을 위한 영상들은 352×288부터 1920×1080까지 다양한 크기의 영상을 사용하였다. 표 1은 본 실험에서 사용한 영상들과 크기를 보여주고 있다.

표 1. 트레이닝에 사용된 20개의 테스트 영상들

Size	Training images
352×288	5 images (Akiyo, Bus, Coastguard, Football, Foreman)
512×512	9 images (Airplane, Baboon, Barbara, Boat, Finger, Girl, Goldhill, Lena, Man)
1280×720	4 images (City, Crew, Jets, Mobcal)
1920×1080	2 images (Bluesky, Kimono)

제안하는 알고리듬과 기존의 알고리듬들과의 성능 비교를 표 2에서 보여주고 있다. 표 2에서 볼 수 있듯이, 제안하는 알고리듬이 가장 큰 PSNR 성능을 보여주고 있다. 특히, 제안하는 방법의 평균 PSNR이 기존의 알고리듬들 중 PSNR 성능이 뛰어난 FDD 알고리듬보다 0.08dB 높으면서 FDD 알고리듬의 CPU time을 2.42%까지 줄였으며, CAD 알고리듬과 비교하면, 0.24dB의 PSNR 향상과 0.59%까지 CPU time을 줄였다. 게다가, 기존의 방법들 중 PSNR 성능이 좋으면서 복잡도까지 낮은 LCID, MELA 알고리듬과 비교할 경우, 제안하는 방법이 두 방법보다 0.11dB 높은 PSNR 수치를 보이면서, 거의 비슷한 속도를 보이고 있다. 그림 8는 MELA, LCID 알고리듬과 제안하는 알고리듬의 8개의 정지영상에 대한 평균 연산량을 보여주고 있다. 연산량은 MELA 와 LCID 알고리듬에 경우, 에지방향 판단하는 부분과 판단 후 픽셀을 보간하는 부분의 연산량을 영상별로 누적시켜 평균을 취하였고, 제안하는 알고리듬의 경우 DoLC를 구하는 연산과 알고리듬별로 에지방향 판단 및 픽셀을 보간하는데 필요한 연산량을 누적시켜 평균을 취하여 구하였다.

그림 8에서 볼 수 있듯이, 제안하는 방법이 MELA 알고리듬보다 더하기 및 빼기 연산(Addition), 곱하기 연산(Multiplication), 절대값 연산(Absolute) 등에서 더 적은 연산량을 필요로하는 것을 볼 수 있고, LCID 알고리듬에 비해서 곱하기 연산의 경우 더 적은 수치를 보이면서, 더하기 및 빼기 연산과 절대값 연산에서는 거의 차이가 없는 수치를 보이고 있다. 위의 결과를 통해 알 수 있듯이, 제안하는 알고리듬이 기존의 알고리듬들보다 PSNR 성능과 계산량 측면에서 우수한 성능을 보이는 것을 확인할 수 있다. 그림 9는 제안하는 알고리듬의 주관적인 측면에서도 성능 평가를 하기 위해 Goldhill 영

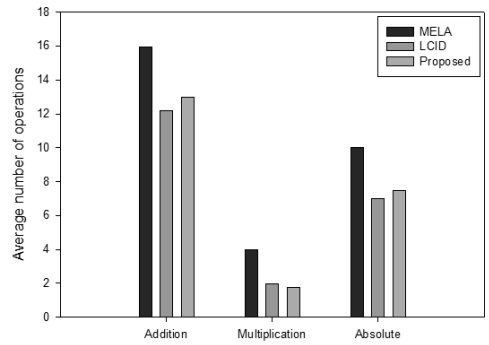


그림 8. 평균 연산량 비교

표 2. 제안하는 알고리듬과 기존의 디인터레이싱 알고리듬과의 성능 비교

		Bluesky	Coast-guard	Goldhill	Raven	Shields	Table tennis	Toys	Zelda	Avg.
ELA	dB	37.24	29.16	32.36	40.03	32.43	29.95	32.60	39.58	34.17
	s	0.25	0.00	0.03	0.11	0.11	0.02	0.11	0.05	0.08
EELA	dB	37.36	29.28	32.70	40.62	32.76	30.41	32.96	40.00	34.51
	s	0.31	0.00	0.03	0.14	0.14	0.02	0.14	0.05	0.10
DOI	dB	37.88	29.81	33.67	42.13	33.54	31.01	33.35	41.31	35.34
	s	5.33	0.66	1.16	1.38	3.33	0.55	0.58	0.44	1.68
NEDD	dB	37.89	29.65	33.46	41.90	33.39	30.74	33.28	41.11	35.17
	s	1.72	0.08	0.20	0.73	0.73	0.08	0.22	0.22	0.50
LCID	dB	38.00	29.75	33.61	41.90	33.47	31.09	33.55	41.15	35.32
	s	0.34	0.02	0.03	0.14	0.14	0.02	0.15	0.05	0.11
MELA	dB	37.98	29.74	33.64	41.98	33.49	31.14	33.34	41.24	35.32
	s	0.42	0.02	0.05	0.17	0.17	0.02	0.15	0.05	0.13
FDD	dB	37.87	29.73	33.71	42.24	33.58	31.06	33.37	41.27	35.35
	s	17.02	1.23	2.58	6.56	8.44	1.13	2.08	2.03	5.13
EMD	dB	37.73	29.74	33.27	41.31	33.27	30.68	33.09	40.58	34.96
	s	1.16	0.05	0.14	0.47	0.47	0.05	0.13	0.14	0.32
ECA	dB	37.59	29.49	32.83	40.66	32.97	30.32	32.76	39.95	34.57
	s	4.21	0.25	0.67	1.97	2.28	0.25	0.53	0.58	1.34
CAD	dB	38.43	29.23	33.34	42.32	33.41	30.08	33.38	41.38	35.19
	s	71.76	3.51	9.02	31.76	31.68	3.42	8.95	8.99	21.14
Prop.	dB	38.03	29.82	33.69	42.20	33.55	31.27	33.53	41.36	35.43
	s	0.39	0.02	0.05	0.17	0.16	0.02	0.15	0.05	0.12

상을 기존의 알고리듬과 제안하는 알고리듬으로 디인터레이싱 한 영상들을 보여주고 있다. 에지 부분을 얼마나 정확하게 복원했는지를 확인하기 위해 정지 영상을 부분적으로 확대하였다. 각각의 그림에서 볼 수 있듯이 제안하는 알고리듬이 기존 알고리듬에 비해 주관적으로 향상된 영상을 보여주는 것을 볼 수 있다.

V. 결 론

본 논문은 공간적 방향의 경향성을 고려하여 알고리듬을 적응적으로 선택하여 적용하는 디인터레이싱 알고리듬을 제안한다. 먼저, 보간 하고자하는 화소 위치에서 DoLC를 구하고, 그 값에 원본 픽셀과 LA, MELA, LCID 알고리듬으로 보간한 픽셀들과

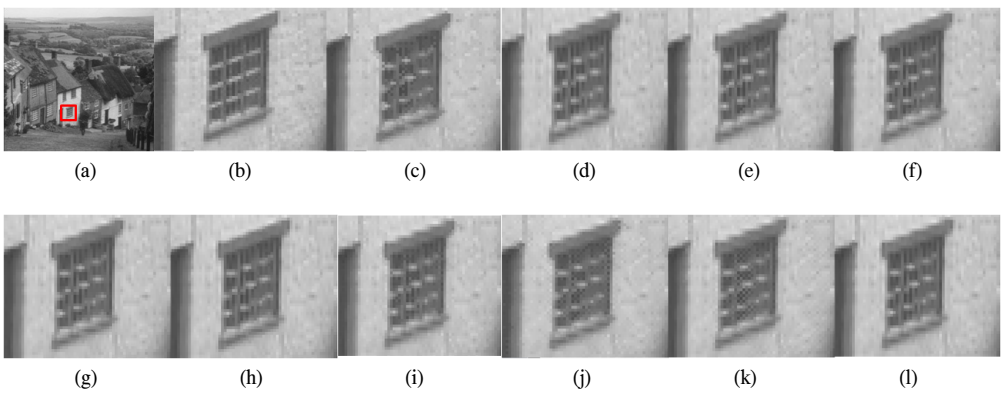


그림 9. "Goldhill" 영상의 주관적 화질 비교. (a) 원본 영상, (b) 원본 블록, (c) EELA, (d) DOI, (e) NEDD, (f) LCID, (g) MELA, (h) FDD, (i) EMD, (j) ECA, (k) CAD, (l) Proposed algorithm

의 차를 구하여 누적시킨다. 여러 특성의 이미지에서 이 작업을 반복하여 정보를 누적하고 세 알고리듬 각각 누적된 값의 평균값을 비교하여 오차 값이 가장 작은 알고리듬을 그 DoLC에 효과적인 알고리듬으로 선택하게 된다. 다양한 특성의 이미지들을 트레이닝해서 LUT를 만들기 때문에 기존의 방법들보다 좋은 화질의 결과를 얻을 수 있고, 계산량이 적은 알고리듬을 적용하기 때문에 연산량이 적어 실시간 시스템에 적합하다. 여러 가지 정지 영상을 대상으로 한 실험 결과를 통해 제안하는 알고리듬이 기존의 알고리듬보다 객관적 및 주관적 화질이 좋다는 것을 볼 수 있다.

참 고 문 헌

〔1〕 Y.-Y. Jung, B.-T. Choi, Y.-J. Park and S.-J. Ko, "An effective de-interlacing technique using motion compensated interpolation," *IEEE Trans. Cons. Elect*., Vol.46, No.3, Aug., 2000.

〔2〕 K. Sugiyama and H. Nakamura, "A method of deinterlacing with motion compensated interpolation," *IEEE Trans. Cons. Elect*., Vol.45, No.3, pp.611-616, Aug., 1999.

〔3〕 C. J. Kuo, C. Liao, and C. C. Lin, "Adaptive interpolation technique for scanning rate conversion," *IEEE Trans. Circuits and Syst. Video Technol.,* Vol.6, No.3, pp.317-321, June, 1996.

〔4〕 T. Chen, H. R. Wu, and Z. H. Yu, "Efficient deinterlacing algorithm using edge-based line average interpolation," *SPIE Opt. Eng*., Vol.39, No.8, pp.2101-2105, Aug. 2000.

〔5〕 H. Yoo and J. Jeong, "Direction-oriented interpolation and its application to de-interlacing," *IEEE Trans. Cons. Elect*., Vol.48, No.4, pp.954-962, Nov., 2002.

〔6〕 M. K. Park, M. G. Kang, K. Nam, and S. G. Oh, "New edge dependent deinterlacing algorithm based on horizontal edge pattern," *IEEE Trans. Cons. Elect*., Vol.49, No.4, pp.1508-1512, Nov., 2003.

〔7〕 P.-Y. Chen and Y.-H. Lai, "A low-complexity interpolation method for deinterlacing," *IEICE Trans. Inf. & Syst*., Vol E90-D, No.2, Feb., 2007.

〔8〕 W. Kim, S. Jin, and J. Jeong, "Novel intra deinterlacing algorithm using content adaptive interpolation," *IEEE Trans. Cons. Elect*., Vol.53, No.3, pp.1036-1043, Aug., 2007.

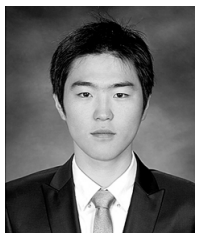
〔9〕 S. Jin, W. Kim, and J. Jeong, "Fine directional de-interlacing algorithm using modified Sobel operation," *IEEE Trans. Cons. Elect*., Vol.54, No.2, pp.857-862, May, 2008.

〔10〕 K. Kang, G. Jeon, and J. Jeong, "A single field interlaced to progressive format conversion using edge map in the image block," in *Proc. IASTED SIP2009*, Innsbruck, Austria, August 17-19, 2009.

〔11〕 T.-H. Tsai and H.-L. Lin, "Design and implementation for deinterlacing using the edge-based correlation adaptive method," *SPIE Journal of Electronic Imaging*, Vol.18, No.1, pp.013014:1-5, Mar., 2009.

〔12〕 S.-J. Park, G. Jeon, and J. Jeong, "Computation -aware algorithm selection approach for interlaced-to-progressive conversion," *SPIE Opt. Eng*., Vol.49, No.5, pp.057005:1-9, May, 2020.

홍 성 민 (Sung-Min Hong) 준회원



2010년 2월 숭실대학교 정보통신전자공학부 학사

2010년 3월~현재 한양대학교 지능형로봇학과 석사과정

<관심분야> Image Processing (compression and enhancement), Fast motion estimation

박 상 준 (Sang-Jun Park) 정회원

2006년 2월 한양대학교 전자컴퓨터공학부 학사

2006년 3월~현재 한양대학교 전자통신컴퓨터공학과 석박사 통합 과정

<관심분야> Image Processing (compression and enhancement), Fast motion estimation, Image resizing

정 제 창 (Jechang Jeong) 정회원

1980년 2월 서울대학교 전자공학과 학사
1982년 2월 KAIST 전기전자공학과 석사
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)
1991년~1995년 삼성전자 멀티미디어 연구소(MPEG, HDTV, 멀티미디어 연구)
1995년~현재 한양대학교 전기통신컴퓨터공학과 교수(영상통신 및 신호처리 연구실)
1998년 11월 27일 과학기술자상 수상
1998년 12월 31일 정보통신부장관상 표창
<관심분야> Image Processing and image compression