

메쉬 블록을 사용한 인쇄 영상에서의 워터마킹

변호준, 김기석, 천인국

순천향대학교 공과대학 정보기술공학부

Watermarking for Printed Image using Mesh Block

Ho-Jun Byun, Gee-Seog Kim, In-Gook Chun

*Division of Information Technology Engineering, Engineering College,
Soonchunhyang University, Asan 336-745, Korea*

요 약

본 논문은 그레이 스케일 인쇄 영상의 저작권을 보호할 수 있는 워터마킹 기법에 관한 연구이다. 기존의 인쇄영상 워터마크 기법 연구는 디터링을 이용하는 까닭에 인쇄 영상의 화질 감소가 발생한다. 본 연구에서는 HVS(human visual system)의 특성을 이용하여 시각적으로 감지할 수 없는 워터마크를 영상 내에 삽입한다. 제안된 알고리즘은 메쉬 블록을 사용하여 그레이 스케일 영상에 워터마크를 삽입하므로 영상의 시각적인 화질 감소가 없을 뿐만 아니라, 프린터와 스캐너에 의한 영상의 기하학적 변형에도 워터마크의 삽입과 추출이 가능하다. 제안된 알고리즘으로 워터마킹하는 경우, 잘라내기(cropping), 회전(rotation), 그리고 스캐너와 프린터의 기기적인 특성에도 강한 워터마크를 삽입하고 추출할 수 있다. 제안된 알고리즘으로 인쇄 영상에 워터마크를 삽입하고 추출하는 실험 결과들은 인쇄 원영상의 화질을 손상하지 않고 기하학적 변형에도 강한 워터마크의 삽입과 추출이 가능함을 보여준다.

1. 서 론

최근 개인용 컴퓨터의 보급 확대와 인터넷의 발달로 인하여 영상, 음성, 비디오 등의 멀티미디어 데이터가 급격히 증가하고 있다[1]. 멀티미디어 표현물의 상업적 가치가 증가되고, 각종 멀티미디어 데이터들이 네트워크와 인터넷을 통해서 분배, 복제되어 이들의 저작권을 보호할 필요성이 증대되고 있다[2]. 또한 디지털 데이터의 불법복본 연구는 한국소프트웨어 진흥원의 ITRC 사업에 의해 수행된 것임.

제 및 변조 등에 대한 문제가 발생하게 되었으며, 이러한 문제를 해결하기 위해 최근에 디지털 데이터의 내부에 정보를 은닉하여 멀티미디어 데이터의 저작권을 보호(copyright protection)할 수 있는 연구가 활발히 진행되고 있다[3]. 워터마킹은 인간의 시각으로는 지각할 수 없는 소유권정보를 디지털 데이터에 삽입함으로써 소유권 인증을 제시하여 주는 방법이다. 디지털 워터마크 기법은 최근 수 년 동안 디지털 멀티미디어 데이터에 대해 올바른 저자를 확인시켜 주는 저작권 보호 방법으로 제안되어 왔다.

위터마킹 기법은 공간 영역 기법과 주파수 영역 기법으로 분류할 수 있다. 공간 영역 기법은 영상에서 선택된 화소의 밝기나 색상을 직접적으로 수정하는 방식으로 위터마킹이 이루어진다. 주파수 영역에서의 위터마킹은 영상을 주파수 영역으로 변환한 후, 변환된 계수 값을 변경하는 방식으로 이루어진다. 이처럼 디지털 영상에 대한 위터마크는 활발히 연구되고 있는 반면에 인쇄 영상에 대한 디지털 위터마킹 연구는 그다지 활발하지 않았다. 최근에 연구된 인쇄 영상의 위터마킹 기법은 디더링(dithering)을 이용하는 까닭에 원래의 영상보다 많은 잡음을 가지게 된다. 많은 잡음을 지닌 디더링 영상은 필연적으로 영상의 화질을 떨어뜨린다. 본 논문은 기존의 디더링을 이용한 위터마킹 알고리즘보다 기하학적 변형에 강하며, 화질저하가 없는 인쇄 영상에서의 위터마킹에 관한 연구이다.

2. 관련 연구

영상을 인쇄할 경우, 영상 데이터가 표현할 수 있는 색상 수를 제한적으로 표현하는 프린터 장비의 제약 때문에, 일반적으로 디더링 기법을 많이 사용한다. 디더링은 CRT 디스플레이, 플라즈마 디스플레이(PDP) 등과 같은 표시장치뿐만 아니라 레이저 프린터, 잉크젯 프린터 등의 장치에서 원영상의 칼라수보다 더 적은 수로 최대한 원영상에 가깝게 표현할 수 있는 기법이다[4]. 그림 1은 오더드(ordered) 디더링의 적용 예이다. 위터마크가 삽입된 영상이라 해도 인쇄나 스캐닝을 하는 경우 디더링에 의해 위터마크 신호가 제대로 보존되지 않는다. 즉 위터마크가 삽입된 영상을 프린트한다면 디더링 기법을 적용함으로써 원영상의 칼라정보와 함께 위터마크 정보까지도 감소되어, 위터마크 신호의 검출이 어려워진다. 결국은 기존 위터마크 신호 내용이 쓸모없게 되므로 인쇄할 경우, 저작권을 보호하는데 문제점이 발생하게 된다. 인쇄 영상의 디지털 위터마킹인 경우에는 디더링, 프린팅, 스캐닝, 복사 등에 견고해야 한다.



그림 1. 오더드 디더링

인쇄 영상 데이터에 대한 디지털 위터마킹에 관한 연구들 중에는, 디더링이 수행된 영상이 원영상보다 많은 잡음을 갖게 되며, 이 디더링 영상이 디지털 위터마킹 관점에서 기밀 정보의 은닉에 적합하다는 성질을 이용한 인쇄 영상 위터마킹 기법이 연구되었다. 특히, Hagit이 제안한 디더링 영상에 대한 위터마킹 알고리즘은 디더링 임계치 패턴을 이용하여 위터마크 신호를 삽입하는 방법이다[5]. 즉, 한 영상에 두 가지의 디더 패턴을 적용하여 위터마크를 삽입하는 방법이다. 이 알고리즘은, 위터마크의 삽입은 간단하지만 디더 패턴의 선택이 잘못될 경우 위터마크 검출이 어려워질 수 있으므로 디더 패턴의 선택에 신중해야하며 검출 과정이 복잡하다는 단점이 있다.

3. 위터마크 삽입 알고리즘

본 연구에서 제안하는 알고리즘은 블록 단위로 위터마크 비트를 영상에 삽입한다. 그레이 스케일 영상의 픽셀 값에 시각적으로 감지할 수 없는 작은 값을 가감하여 양자화를 수행한다. 픽셀 단위로 양자화를 하는 경우, 확대·축소, 회전, 크롭핑과 같은 기하학적 변형에 위터마크가 쉽게 왜곡된다. 이런 까닭에, 제안된 알고리즘은 사각 블록 안에 있는 모든 픽셀에 한 비트의 위터마크를 양자화 한다. 블록 단위로 양자화하는 경우, 프린터나 스캐너의 기하학적 변형에도 위터마크를 추출할 수 있다. 그림 2는 구현된 시스템의 구성도이다.

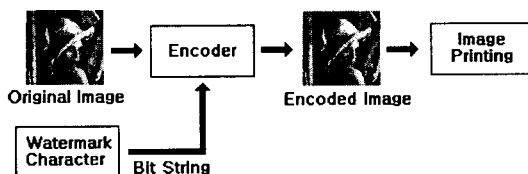


그림 2. 위터마크 삽입 다이어그램

3.1 프린터의 광학적 특성

본 절에서는 프린터의 광학적 특성에 관하여 알아본다. 프린터는 디지털 영상의 모든 색상을 표현할 수 없다. 256 그레이 스케일 영상을 인쇄하는 경우, 흰색(255)이나 검정색(0) 주위의 색상들은 모두 흰색 또는 검정색으로 인쇄된다. 즉, 픽셀 값이 약 220보다 큰 경우에는 모두 흰색으로 인쇄되고, 반대로 약 35보다 작은 값은 모두 검정색으로 인쇄된다. 이러한 프린터의 광학적 특성 때문에, 블록 단위로 워터마크를 삽입하여 워터마크를 추출할 경우, 주위의 픽셀 값들도 참조할 수 있도록 해야 한다. 만약 주위의 픽셀 값들이 모두 흰색이나 검정색에 가까운 경우에는 현 블록에 워터마크를 삽입하지 않고 다음 블록으로 이동하여 워터마크를 삽입하도록 구현하였다.

3.2 픽셀의 양자화

본 절에서는 영상의 공간 영역에 해당하는 픽셀에 워터마크를 삽입하는 알고리즘에 관하여 논한다. 픽셀 값은 영상의 화질에 직접적인 영향을 주므로 HVS로 감지할 수 없는 작은 값의 변경으로 양자화를 수행해야 한다. 그림 3에서와 같이 픽셀 값이 83인 경우에, 워터마크 비트가 1이면 83에 가장 가까운 10의 배수를 현 픽셀의 값으로 변경하고, 워터마크 비트가 0이면 10의 배수를 제외하고 83에 가까운 5의 배수를 현 픽셀의 값으로 변경한다.

Pixel Value : 83
 Watermark Bit : 1 → 10의 배수 : 80
 Watermark Bit : 0 → 5의 배수 : 85

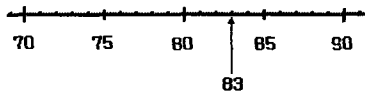


그림 3. 픽셀의 양자화 알고리즘

그림 3과 같은 양자화를 수행하는 경우, 256 그레이 스케일 영상의 화질에 영향이 없이 워터마크 비트의 삽입이 가능하다. 그림 3과 같은 양자화 알고리즘으로 워터마크를 삽입하는 알고리즘은 그림 4와 같다.

```

int Remain = Curr_Pixel % 10;
if (Watermark_Bit == 1)
{
    Curr_Pixel = Curr_Pixel - Remain;
    if (Remain > 5)
        Curr_Pixel += 10;
}
else // Case Of (Watermark Bit = 0)
{
    Curr_Pixel = Curr_Pixel - Remain + 5;
}
    
```

그림 4. 워터마크 삽입 알고리즘

3.3 워터마크 인코딩 블록

3.2에서와 같이 픽셀 단위로 워터마크(비트)를 삽입하는 경우, 워터마크 삽입 대상이 되는 영상이 프린터나 스캐너에 의해 기하학적으로 변형되는 경우, 워터마크의 추출이 불가능해진다. 본 연구에서는 워터마크가 삽입된 영상에 기하학적인 변형이 가해진 경우에도 워터마크를 추출할 수 있도록 하기 위하여 블록 단위로 워터마크를 삽입하였다. 그림 5는 254×255인 그레이 스케일 영상에 삽입된 워터마크를 시각적으로 표시한 것이며 흰색 부분은 워터마크 비트가 1인 경우이고 검정색인 부분은 0인 경우이다. 그림 5에서 워터마크가 삽입되지 않은 부분은 워터마크 블록의 크기에 맞춰 워터마크를 삽입하기 때문에 발생하는 여백이다.

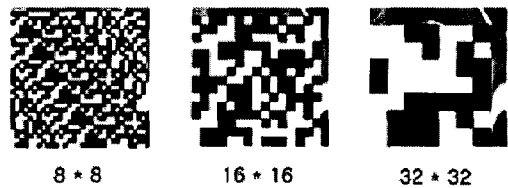


그림 5. 워터마크 블록의 예

블록 단위로 워터마크를 삽입함으로써 확대·축소, 회전, 영상 왜곡과 같은 변형에 강인성을 갖는다. 블록의 크기는 저작권자가 임의로 크기를 선정할 수 있다. 그림 5는 가로·세로의 픽셀 수를 다르게 했을 경우의 블록의 크기를 보여준다. 블록의 크기가 크면 워터마크 추출 신뢰도는 높아지지만 삽입 가능한 워터마크의 양은 감소한다.

4. 워터마크 추출 알고리즘

본 장에서는 제안된 알고리즘으로 삽입된 워터마

크를 추출하는 방법에 대하여 논한다. 인쇄 영상은 프린터나 스캐너의 광학적인 특성을 받게 되므로 이러한 특성에 대한 정보를 디코더가 가지고 있어야 한다. 구현된 시스템에서는 프린터와 스캐너의 광학적 특성을 파악하기 위해 처음 두 블록에 워터마크 비트가 '1'일 때와 '0'일 때의 양자화를 수행하여 픽셀 값을 변경하였다. 디코더는 첫 번째 블록을 조사하여 워터마크 비트가 '1'일 때의 광학적 특성을 조사하고 두 번째 블록을 조사하여 워터마크 비트가 '0'일 때의 광학적 특성을 파악하게 된다. 파악된 특성 값을 가지고 각 블록에 어떤 워터마크 비트가 삽입되었는지 알아낸다. 그림 6은 구현된 시스템의 워터마크 추출 다이어그램이다.



그림 6. 워터마크 추출 다이어그램

4.1 스캐너의 광학적 특성

본 절에서는 스캐너의 광학적 특성에 관하여 알아본다. 스캐너 또한 프린터와 마찬가지로 광학적인 특성을 가지고 있다. 스캐닝을 하는 동안 인쇄된 원영상에 없던 잡영이 영상에 첨가될 수도 있으며 광학적인 왜곡이 발생할 수도 있다. 본 연구에서는 이러한 프린터와 스캐너의 광학적인 특성을 고려한 워터마크 추출 시스템을 구현하였다. 그림 7은 프린터와 스캐너의 특성(capacity)을 고려한 워터마크 추출 알고리즘이다. 워터마크를 추출하는 경우, 프린터와 스캐너의 특성을 고려해 주지 않는 경우, 워터마크 비트를 추출할 수 없다. 제안된 워터마크 삽입 알고리즘에서는 워터마크 비트가 '1'인 경우와 '0'인 경우의 양자화 값을 삽입 영상의 처음 두 블록에 삽입하여 디코더에서 프린터와 스캐너의 특성 값을 파악할 수 있도록 하였다. 제안된 알고리즘의 제약점에 관해서는 5장에서 논한다.

```

int Remain = Curr_Pixel % 10 ±
Capacity;
if (Remain == NULL)
{
    Watermark_Bit = 1;
}
else
{
    Watermark_Bit = 0;
}
  
```

그림 7. 워터마크 추출 알고리즘

4.2 기하학적으로 변형된 영상의 디코딩

그림 8에서와 같이 3×3 메쉬 블록을 사용하는 경우, 약 5° 정도의 회전에도 워터마크를 추출할 수 있으며, 블록의 크기에 따라 크랩핑에도 강한 특성을 보인다. 그림 8은 메쉬 블록이 3×3이 아니라 2×2인 경우일 때에는 약 45° 정도 까지의 회전에 대하여 포용력(tolerance)을 보일 것이다. 게다가, 메쉬 블록을 사용함으로써 프린팅과 스캐닝 때문에 발생하는 확대·축소된 경우에도 원래의 영상 크기로 스케일링한 후, 워터마크를 추출할 수 있다.

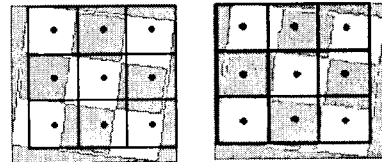


그림 8. 메쉬 블록의 적용 예

제안된 알고리즘은 프린터와 스캐너의 광학적인 왜곡에 대한 강인성을 갖도록 하기 위해 인쇄 영상의 처음 두 개의 워터마크 블록에 '1'과 '0'으로 양자화했으며, 기하학적인 왜곡에 강인성을 갖도록 하기 위해 메쉬 블록을 사용하였다. 공격자의 인위적인 잘라내기와 왜곡을 가하는 경우에는 워터마크의 추출이 불가능해진다. 하지만, 이러한 인위적인 왜곡을 가하는 경우에는 프린팅과 스캐닝 때문에 영상의 화질이 낮아진 상태이므로 영상의 품질이 더욱 낮아져 콘텐츠로서의 가치가 사라진다.

5. 실험환경 및 결과

본 논문은 실험환경 Pentium III 800Mhz 프로세

서와 Visual C++ 6.0으로 워터마킹 시스템을 구현하였고 그레이 스케일 영상을 대상으로 실험한다. 제안된 알고리즘으로 양자화 하는 경우 화질의 저하가 없음을 증명하기 위해서 8×8, 16×16, 32×32의 워터마크 블록으로 나누어진 이미지의 화질을 비교한다. 사용된 워터마크는 "SCH.Univ"라는 문자열을 사용하였으며, 워터마크 인코더는 입력된 문자열에서 비트들을 추출하여 각각의 워터마크 블록에 삽입한다. 워터마크가 삽입된 각각의 영상을 프린트하고 다시 스캔하여 워터마크를 추출하였다. 실험 영상에 사용된 워터마크 블록의 크기는 각각 8×8, 16×16, 32×32, 64×64 크기로 하였다. 본 실험에서는 삽입된 워터마크의 비가시성(invisibility)과 회전(rotation)의 공격에 대한 강인성과 확대·축소와 같은 스케일링 변환에 대한 워터마크 추출 신뢰도를 측정하였다.

5.1 영상의 화질 분석

그림 9는 원영상과 서로 다른 워터마크 블록을 이용하여 워터마크를 삽입하였을때의 이미지와 비교한 것이다.

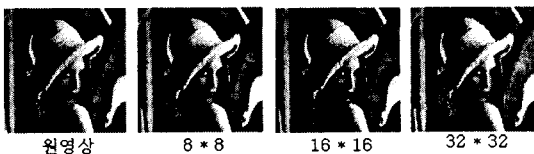


그림 9. 원영상과 여러 블록으로 삽입된 영상

육안으로 크게 차이를 보이지 않고 있는데, 이는 HVS의 성질을 이용하여 양자화 하여 미세한 픽셀값의 변화에 대해서 인간이 인지할 수 없다는 것을 이용한 것이다.

5.2 회전과 신축에 대한 강인성 실험

본 실험에서 사용된 영상은 256×256 크기의 그레이 스케일 영상이며, 사용된 워터마크 블록의 크기는 8×8, 16×16, 32×32, 64×64이다. "SCH.Univ" 라는 문자열을 워터마크로 삽입하였을때 회전 변환과 신축(축소, 확대)으로부터 일만분의 공격을 하였을때 삽입된 워터마크 문자열이 확실히 추출 될 수 있는지 스캔한 영상으로 실험하였다.

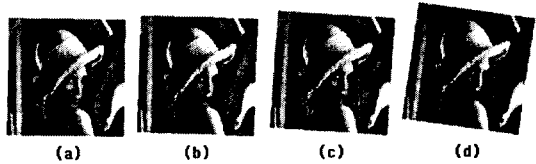


그림 10. (a) 1° 회전된 8×8영상 (b) 2° 회전된 16×16영상 (c) 4° 회전된 32×32영상 (d) 7° 회전된 64×64영상

표 1. 회전에 대한 워터마크 추출

워터마크 블록	비트수	허용 최대각
8×8	1024 bit	1°
16×16	256 bit	2°
32×32	64 bit	4°
64×64	16 bit	7°

위의 표1에서 최대각은 워터마크가 삽입된 영상을 회전하였을때 워터마크가 추출 될 수 있는 각의 최대치를 말하고, 비트수는 문자가 들어 갈 수 있는 크기를 말한다. 그림 10은 표 1의 각각의 실험 결과에 대한 이미지이다. 사용되는 워터마크 블록이 작으면 작을 수록 회전에 대하여 약한 모습을 보여주었다. 블록이 커지면 단위 면적당 차지하는 면적이 커져 회전한 이미지에 대한 워터마크를 추출하기 용이하기 때문이다.

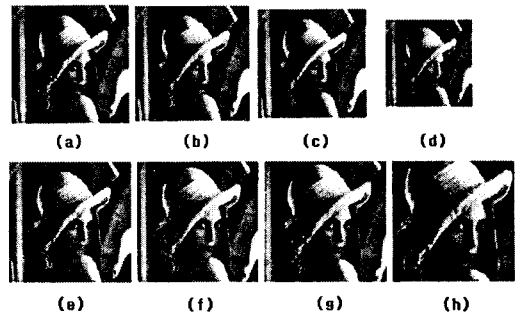


그림 11. (a) 96% 축소된 8×8영상 (b) 94% 축소된 16×16영상 (c) 88% 축소된 32×32영상 (d) 70% 축소된 영상 (e) 102% 확대된 8×8영상 (f) 110% 확대된 16×16영상 (g) 116% 확대된 32×32영상 (h) 149% 확대된 영상

표 2. 스케일링에 대한 워터마크 추출

워터마크 블록	허용 축소율(%)	허용 확대율(%)
8×8	96	102
16×16	94	110
32×32	88	116
64×64	70	149

표 2는 스케일링의 공격에 대한 워터마크 추출에 대한 결과 표로서 그림 11은 이에 대한 결과를 그림으로 나타내고있다. 그림 11의 확대 이미지는 크롭핑의 공격을 고려한 실험 이미지 이다. 원 영상을 100%로 하였을때 표에 나온 퍼센트 만큼 축소와 확대를 시켜보았다. 그 결과 워터마크 블록이 가장 큰 64×64 블록이 축소와 확대에 대하여 강하였고, 워터마크 블록이 클수록 회전과 신축에 대해 강함을 알 수 있었다. 하지만 실험 결과로 볼때 워터마크 블록이 크면 클수록 블록 비트수는 감소하여 많은 데이터를 삽입할 수 없었다.

6. 결론 및 추후과제

본 논문에서는 메쉬 블록을 이용한 프린트 워터마크를 위한 알고리즘을 구현하였다. HVS의 특성을 이용하여 시각적으로 감지하기 어려운 워터마크를 삽입하였고, 제안된 알고리즘은 워터마크 블록을 이용하여 그레이 스케일 영상에 워터마크를 삽입하므로 영상의 질이 저하되지 않을뿐 아니라, 영상의 회전과 신축에도 워터마크 추출이 가능하였다. 앞으로의 연구로 실제 프린트와 스캐너 실험을 통한 워터마킹 삽입 추출을 실험할 예정이고 또한 워터마킹 블록의 크기에 따른 검출 신뢰성에 관한 연구가 필요하리라 본다.

참고 문헌

[1] F. A. P. Petitcolas, R. j. Anderson, and M. G. Kuhn, "Information hiding - A survey," Proc. IEEE, vol. 87, no. 7, pp. 1062-1078, Jul. 1999.

[2] M. D. Swanson, M. Kobayashi, and A. Tewfik, "Multimedia Data-Embedding and Watermarking Technologies," Proc. of IEEE,

vol. 86, no.6, pp. 1064-1097, 1998

[3] M. D. Swanson, B. Zhu, and A. Tewfik, "Data Hiding for Video and Other Applications," IEEE Int. Conf. on Image Processing, vol. 3, pp. 211-214, 1997

[4] R. Ulichney, "Dithering with Blue Noise". Proceedings of the IEEE, vol. 76(1), pp. 56-79, 1998.

[5] Hagit Z. Hel-Or, "Copyright labeling of printed images", IEEE International Conference on Image Processing, vol.3, pp.702-705, 2000.