한국산학기술학회논문지
Vol. 7, No. 4, pp. 656-664, 2006

LUT(Look-Up Table)을 사용한 인쇄 영상의 워터마킹

천인국[1*]

Printable Image Watermarking Based on Look-Up Table

In-Gook Chun[1*]

요 약 본 논문에서는 하프톤 영상을 LUT(Look-Up Table)를 이용하여 워터마킹하는 새로운 알고리즘을 제안한다. 제안된 방법에서는 워터마크 비트가 하프톤 영상내의 랜덤한 위치의 픽셀에 저장된다. 하프톤 영상의 픽셀값은 그레이스케일 픽셀값과 이웃한 하프톤 픽셀값으로 인덱싱되는 LUT 항목이 된다. LUT는 그레이스케일 영상과 대응되는 하프톤 영상의 집합에 의하여 학습된다. 본 방법의 장점은 다른 방법들에 비하여 상당히 빠른 속도로 워터마킹할 수 있다는 것이다. 따라서 본 알고리즘은 프린터안에 내장될 수 있다. 실제로 스캔된 영상들을 이용한 실험에서 제안된 알고리즘이 가시적인 왜곡 없이 많은 양의 워터마크 데이터를 영상 안에 감출 수 있으며 DHED와 비교하여 품질은 비슷하고 실행시간은 단축된다.

Abstract In this paper, we introduce a new LUT based watermarking method for a halftone image. Watermark bits are hidden at pseudo-random locations of halftone image in the proposed method. The pixel values of the halftone image are determined from the LUT entry indexed by both the neighborhood halftone pixels and current grayscale value. The LUT is trained by a set of grayscale images and corresponding halftone images. Advantage of the LUT method is that it can be executed very fast compared with other watermarking methods for a halftone image. Therefore, the algorithm can be embedded in a printer. Experiments for real scanned images showed that the method is a feasible method to hide the large amount of data within a halftone image without noticeable distortion and comparing to the DHED method, is almost same in quality but significantly shorten in processing time.

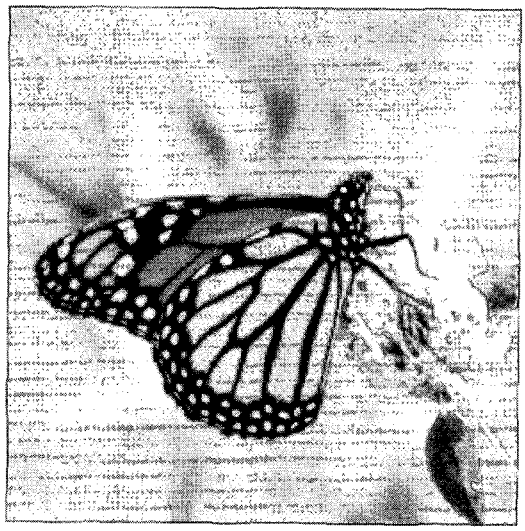
Key Words : 워터마킹, 워터마크, 하프톤 영상, Look-Up Table, 영상처리

1. 서론

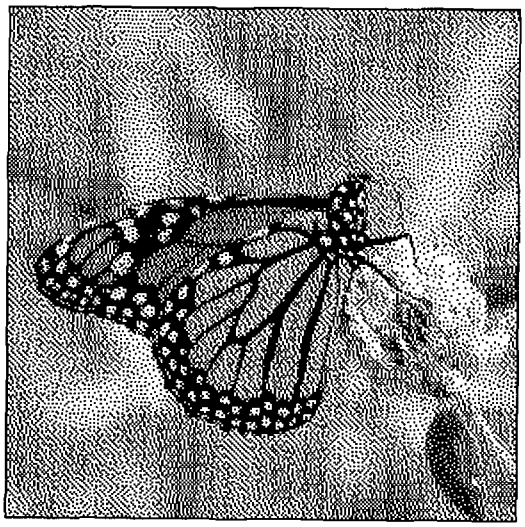
디지털 영상 워터마킹(digital image watermarking)은 영상 안에 저작권 정보를 보이지 않게 삽입하여 영상의 저작권을 보호하기 위한 기법이다. 좋은 워터마킹 기법은 영상의 품질에 최소한으로 영향을 주면서 충분한 저작권 정보를 저장할 수 있어야 한다. 지금까지는 주로 온라인 영상, 즉 온라인상에서 디지털 파일형태로 존재하는 영상에 대한 워터마킹 연구가 주로 이루어져왔다. 이분야에서 가장 많이 사용되는 워터마킹 방법은 영상의 공간 영역이나 주파수 영역에서 워터마크를 삽입하는 방법이다. 그러나 인쇄에 주로 사용되는 하프톤(halftone) 영상에 대한 워터마킹 연구는 상대적으로 미진한 편이다. 하프토닝(halftoning)은 그레이스케일 영상을 2가지의 색상 즉 흰색과 검정색만을 이용하여 표현하는 방법이다. 하프토닝은 주로 프린터에서 인쇄과정에서 사용되는데 왜냐하면 프린터들은 대개 제한된 숫자의 색상만을 가지고 있기 때문이다. 현재 흑백이나 컬러를 막론하고 거의 모든 프린터에서 하프톤 기법을 이용하여 영상을 인쇄한다.

현재 사용되고 있는 하프토닝에는 여러 가지가 있다[1][2]. 가장 많이 사용되는 하프톤 기법은 오더드 디더링(ordered dithering), 오류 확산법(error diffusion), 반복적 기법(iterative halftoning)등의 3가지이다. [그림 1]에 대표적인 하프톤 기법인 오류 확산법을 이용하여 생성한 하프톤 영상을 나타내었다.

본 연구과제는 2004학년도 순천향대학교 산업기술연구소 학술연구조성비 일반연구과제로 지원을 받아 수행하였음
[1]순천향대학교 컴퓨터학부
*교신저자: 천인국(chunik@sch.ac.kr)



(a)



(b)

그림 1. 하프토닝의 개념 (a) 원래의 그레이스케일 영상 (b) 오류확산법으로 생성한 하프톤 영상

오더드 디더링에서는 먼저 입력 영상이 공간적으로 주기적인 스크린을 이용하여 이진화 된다. 오더드 디더링은 간단한 점기반 연산만을 이용하기 때문에 계산이 간단하다는 장점이 있으나 주기적인 패턴이 나타나는 단점이 있다. 오류 확산법에서는 그레이스케일 픽셀값과 하프톤 픽셀값간의 오류가 이웃 픽셀로 전달되어 보다 정확한 하프토닝이 가능하게 한다. 그러나 알고리즘이 병렬적이지 못하고 연산 시간이 오래 걸리는 단점이 있다. 최근에 LUT를 이용한 새로운 하프토닝 방법이 제안되었다. LUT 하프토닝 방법의 장점은 복잡한 계산을 필요로 하지 않는다는 점이다. 단순히 LUT에 대한 간단한 메모리 접근만을 이용하여 하프토닝하는 방법이다. 또한 LUT는 여러가지 다양한 하프토닝 알고리즘을 이용하여 학습이 가능하다.

본 논문에서는 LUT(Look-Up Table)를 이용한 하프톤 영상에 대한 워터마킹 기법을 제안하였다. 현재의 대부분의 워터마킹 알고리즘들은 그레이스케일 영상을 가정하고 만들어진 것이므로 하프톤 영상에는 적용할 수 없다. 그동안 하프톤 영상을 위한 몇 가지의 워터마킹 알고리즘들이 제안되었다. 대표적으로 2가지의 알고리즘을 들 수 있다. 첫번째 알고리즘은 오더드 디더링 방법에서 하나가 아닌 두개의 서로 다른 하프토닝 스크린을 사용하는 방법으로 워터마크 비트값에 따라 서로 다른 하프토닝 스크린을 이용하여 하프토닝한다 [3-5]. 여기서의 주된 문제점은 왜곡이 적고 워터마크 비트 추출이 쉬운 최적의 결과를 얻기 위하여 어떤 스크린 쌍을 사용하여야 하는지를 결정하는 것이 상당히 어렵다는 것이다. 두번째 알고리즘은 Ming과 Oscar에 의하여 제안된 방법으로 워터마크 비트를 직접 하프톤 영상의 일부 픽셀값으로 저장하는 방법이다. Ming과 Oscar는 주로 이 방식을 이용하여 에러확산을 이용하는 하프토닝 기법에 대한 여러가지 워터마킹 기법을 발표하였다 [6-9]. 그들이 발표한 방법 중 가장 대표적인 것이 DHED(Data Hiding Error Diffusion)로서 이는 워터마킹과 오류확산을 결합시킨 것이다. 즉 하프톤 영상의 일부 픽셀을 워터마크 정보에 따라 강제적으로 0 또는 1로 만들고 이로 인한 하프토닝 오류를 다음 하프톤 픽셀을 생성하는데 참조하는 방식이다. 이 방식은 비교적 구현이 쉬운 장점이 있지만 하프토닝 오류가 스캔 순서에 따른 다음 픽셀들에 의해서만 보정되는 약점을 가진다. 따라서 하프톤 오류가 전체 이웃 픽셀에 의해서 보정이 되지 않기 때문에 왜곡이 상당부분 남아있게 된다.

본 논문에서 제안된 방법은 크게 나누면 두번째 방법에 속한다. 즉 하프톤 영상의 일부 픽셀에 워터마크 비트들을 저장한다. 하지만 이들 워터마크 비트들로 인한 영상의 왜곡을 최소화하기 위하여 다음 하프톤 픽셀 값을 선택할 때 LUT를 사용한다. LUT를 사용하게 되면 연산 자체가 아주 최소한의 시간(한번의 메모리 접근)만을 필요로 하기 때문에 큰 장점이 된다. 본 연구에서는 또한 워터마크 패턴 자체를 주기적으로 하여서 영상의 절단이나 기하학적인 변형에 강인하게 하였다. 워터마크 추출 알고리즘은 여러 단계의 영상처리과정을 거치는 완전한 블라인드 워터마킹 알고리즘으로서 난수 발생기의 시드값 이외에는 아무런 다른 정보를 요구하지 않는다. 본 논문에서는 실제로 프린터와 스캐너를 이용하여 인쇄, 스캔

된 영상에 대하여 실험을 하고 그 결과를 제시하였다. 신분증의 회사 로그를 하프톤 영상으로 인쇄하고 여기에 사원 번호를 워터마크로 삽입하여 실험하였다.

2. 워터마크 삽입

본 논문에서의 워터마크는 이진 비트들의 집합으로 가정한다. 따라서 하프톤 영상과 워터마크 정보가 모두 이진이기 때문에 하프톤 영상의 픽셀값에 워터마크 정보를 숨길 수 있다. 따라서 먼저 특정한 비밀키(secret key)를 시드값(seed value)으로 사용하여 난수발생기(random number generator)로부터 난수를 발생하고 이 난수를 이용하여 하프톤 영상내의 랜덤한 위치를 지정하고 여기에 워터마크 비트들을 저장하였다. [그림 2]와 같이 워터마크 비트가 0이냐 1이냐에 따라 하프톤 영상의 특정 픽셀값이 0 또는 1로 강제적으로 설정되게 된다. 만약 이 시드값이 워터마크 추출자에게 전달된다면 추출자는 단순히 시드값으로 동일한 랜덤한 위치들을 발생시켜서 이 위치에 해당되는 픽셀의 값을 읽으면 워터마크를 추출할 수 있다.

여기서 해결하여야할 주된 문제는 삽입된 워터마크로 인하여 야기되는 하프톤 영상의 왜곡에 있다. 즉 하프톤 영상의 일부 픽셀에 워터마크 정보를 저장하다보면 워터마크 픽셀들이 pepper-and-salt 타입의 잡음처럼 보이게 된다. 따라서 이러한 영상의 왜곡을 어떤 방법으로 줄일 것이냐 하는 것이 하프톤 영상 워터마킹의 주된 연구 주제가 된다. 본 논문에서는 이러한 잡음을 하프톤 과정에서 고려하여 최소한의 왜곡만을 가지는 하프톤 영상을 원 영상으로부터 생성하고자 하였다. 제안된 알고리즘에

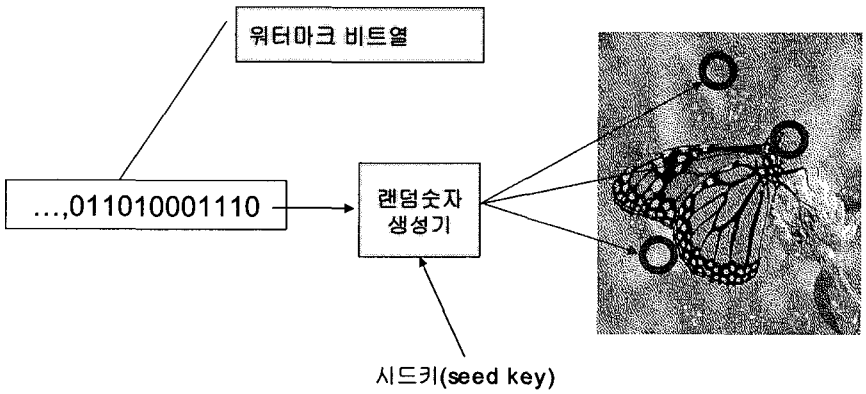


그림 2. 워터마크의 삽입: 워터마크는 시드값에 의해 생성되는 랜덤한 위치의 픽셀값으로 저장된다.

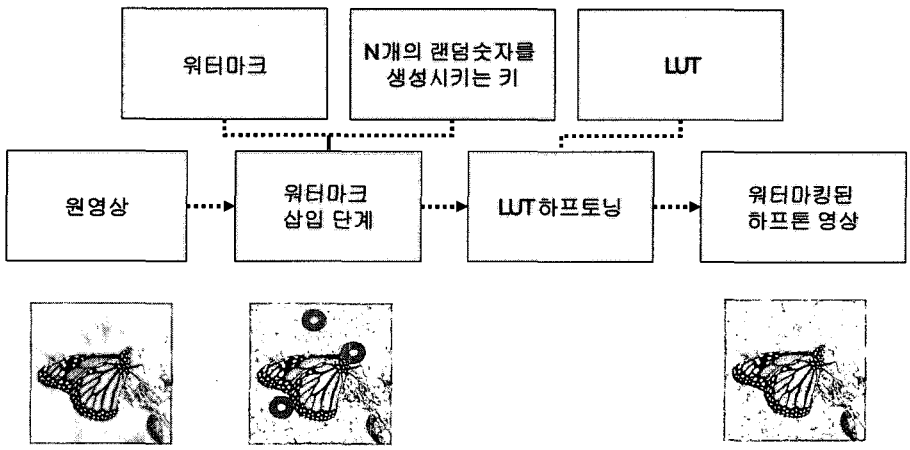


그림 3. 워터마크 삽입 알고리즘

서는 LUT에서 이웃한 픽셀들의 배치를 고려하여 왜곡을 줄이는 최적의 픽셀값을 선택함으로써 이러한 왜곡을 최소화한다. 본 논문에서 제안한 하프톤 영상의 워터마크 삽입 알고리즘의 전체 블록도는 [그림 3]과 같다.

먼저 LUT를 이용한 하프토닝은 [그림 4]와 같이 진행된다. 그레이스케일 영상인 입력영상은 LUT를 이용하여 한 픽셀씩 차례대로 하프토닝되어 이진영상을 생성한다. LUT의 입력은 현재 위치에서의 그레이스케일값과 이미 하프토닝된 주위 픽셀들의 값이다. 만약 [그림 4]에서처럼 현재 픽셀의 값이 35이고 하프토닝이 끝난 주위 픽셀들의 값이 1,0,0,1이라면 이들 조건에 대응되는 출력픽셀값을 LUT에서 찾는다. 즉 하프토닝은 그 위치에서의 그레이스케일 값과 이웃한 주위 픽셀들의 값을 LUT에서 참조하여 이루어진다. 이웃한 픽셀들중에서 아직 하프토닝되지 않은 픽셀들은 참고할 수 없다. LUT는 다른 하프토닝 방법을 이용하여 미리 하프토닝된 영상들을 이용하여 학습시킨다. [그림 4]에서는 편의상 3×3의 윈도우로 표현되었지만 실제로는 [그림 5]와 같은 윈도우가 사용된다.

워터마크의 삽입은 하프토닝과 동시에 이루어진다. 워터마크 데이터는 이진 비트열로 가정한다. 먼저 워터마크 삽입자와 추출자만이 알고 있는 비밀키를 이용하여 N개의 랜덤한 위치를 생성한다. 워터마크데이터의 각각의 비트들은 바로 이 N개의 랜덤한 위치에 저장된다. 하프토닝 과정에서 현재 변환중인 픽셀 위치가 워터마크 비트가 삽입되어야 하는 위치이면 하프톤 영상의 픽셀값은 워터마크 비트가 된다. 저장된다. 그렇지 않다면 LUT를 사용하여 현재 픽셀의 그레이 스케일값과 하프톤 영상에서의 주위 픽셀들의 값들이 LUT에 입력되고 LUT의 출력이 바로 현재 위치에서의 하프톤 영상의 픽셀값이 된다. 본 연구에서는 더욱 품질좋은 하프톤 영상을 생성하기 위하여 1차 하프토닝 과정이 끝나면 2차 하프토닝 과정을 실행하게 된다. 2차 하프토닝에서는 8방향의 주위 픽셀들을 모두 사용하여 다시 한번 하프톤 영상을 생성하게 된다. 1차 하프토닝과 2차 하프토닝에서 사용하는 윈도우는 [그림 5]과 같다.

	15			14	
	11	9	6	8	
13	7	3	2	4	12
10	5	1	O		

(a) 1차 하프토닝 윈도우

		11		
13	5	3	7	
9	1	O	2	10
	6	4	8	14
		12		

(b) 2차 하프토닝 윈도우

그림 5. 하프토닝에서 사용되는 LUT 윈도우

특정 위치에서의 하프톤 픽셀값을 결정하기 위하여 윈도우 안의 이미 결정된 주위 픽셀들의 하프톤 픽셀값과 현재 위치에서의 그레이스케일 값이 사용된다. 윈도우 디자인 단계에서 어떤 주위 픽셀의 값이 포함될 것인지가 결정된다. [그림 5]에서 "O"는 하프톤 값을 결정하려고 하는 픽셀을 나타내고 다른 숫자들은 주위 픽셀들을 나타낸다. 1차 하프토닝에서는 15개의 주위 픽셀을 참조하게 되고 2차 하프토닝에서는 14개의 주위 픽셀을 참조하게 된다. 2차 하프토닝의 경우, 1차 하프토닝을 통하여 이미 모든 하프톤 픽셀들이 결정되어 있으므로 상하좌우

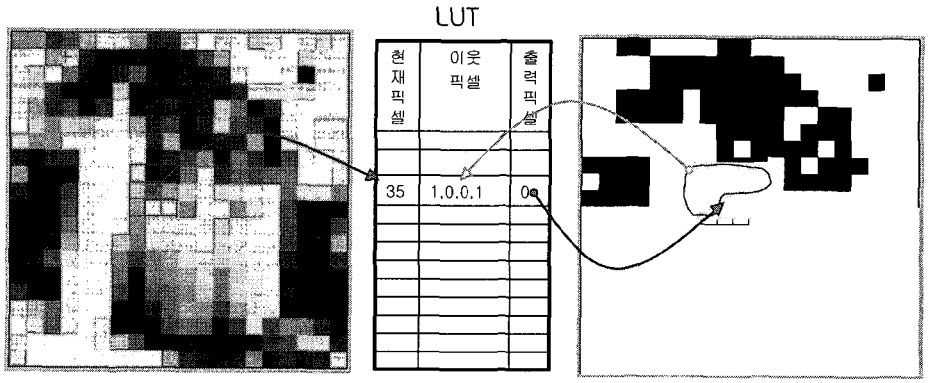


그림 4. LUT를 이용한 하프토닝

의 모든 주위픽셀의 값을 이용할 수 있다.

여기서 윈도우 안에 N개의 픽셀이 있다고 가정해보자. 이들 픽셀의 하프톤 값을 $p_0, p_1, ..., p_{n-1}$이라고 하프톤 값을 결정하려고 하는 픽셀의 그레이스케일 값을 c라고 하자. 이들 값들이 변화할 수 있는 범위를 고려하면 $p_i \in \{0,1\}$이고 $c \in \{0,1,...,2^8-1\}$이므로 총 $2^n 2^8$개의 서로 다른 픽셀 패턴이 존재함을 알 수 있다. 하프톤 영상은 이진 영상이므로 LUT가 이들 패턴에 대하여 반환해야하는 값은 이진값으로 $T(p_0, p_1, ..., p_{n-1}, c)$라고 하자.

LUT를 학습시키는 과정에 대하여 살펴보자. LUT는 샘플 영상들을 다른 하프톤 방법으로 먼저 하프토닝한 후에 이들을 이용하여 LUT를 학습시키게 된다. 먼저 각각의 패턴에 대하여 기댓값을 구한 다음에 이들 기댓값을 이진화하여 LUT에 저장된다. 먼저 패턴 $(p_0, p_1, ..., p_{n-1}, c)$의 발생횟수를 $K_{p0,p1,..,pn-1,c}$라 하자. 그리고 각각의 하프톤 값을 다음과 같다고 가정하자.

$$h_{p0,p1,...,pn-1,c,i} \quad \text{for } i = 0, 1, ..., K_{p0,p1,...,pn-1,c} \tag{1}$$

만약 $K_{p0,p1,...,pn-1,c} > 0$이면 이 패턴에 대한 LUT값은 하프톤 값들의 평균에 가장 근접하는 이진값이 된다. 즉 다음과 같다.

$$T(p_0 p_1 ... p_{N-1} c) = \begin{cases} 1 & \text{if} \quad m_{p0p1...pN-1c} \geq 0.5 \\ 0 & \text{if} \quad m_{p0p1...pN-1c} < 0.5 \end{cases} \tag{2}$$

여기서

$$m_{p0p1...pN-1c} = \frac{\sum_{i=0}^{K_{p0p1...pN-1c}-1} h_{p0p1...pN-1c,i}}{K_{p0p1...pN-1c}} \tag{3}$$

이다.

여기서 문제가 되는 것은 학습시에 존재하지 않았던 패턴들에 대해서는 어떻게 하프톤 값을 정하느냐 하는 것이다. 여러 가지의 방법이 존재하는데 비교적 좋은 성능을 보이는 방법은 best linear estimator를 사용하는 것이다 [10]. 만약 현재 하프톤값을 결정하려는 픽셀이 워터마크 비트를 저장해야 된다면 이러한 절차를 생략하게 된다. 즉 워터마크 비트의 값에 따라서 하프톤값이 결정된다.

3. 워터마크 추출

워터마크의 추출은 워터마크를 담고 있는 하프톤 영상을 프린터로 인쇄한 다음, 이 인쇄된 하프톤 영상을 다시 스캐너로 스캔하여 여기에서 워터마크를 추출하는 것으

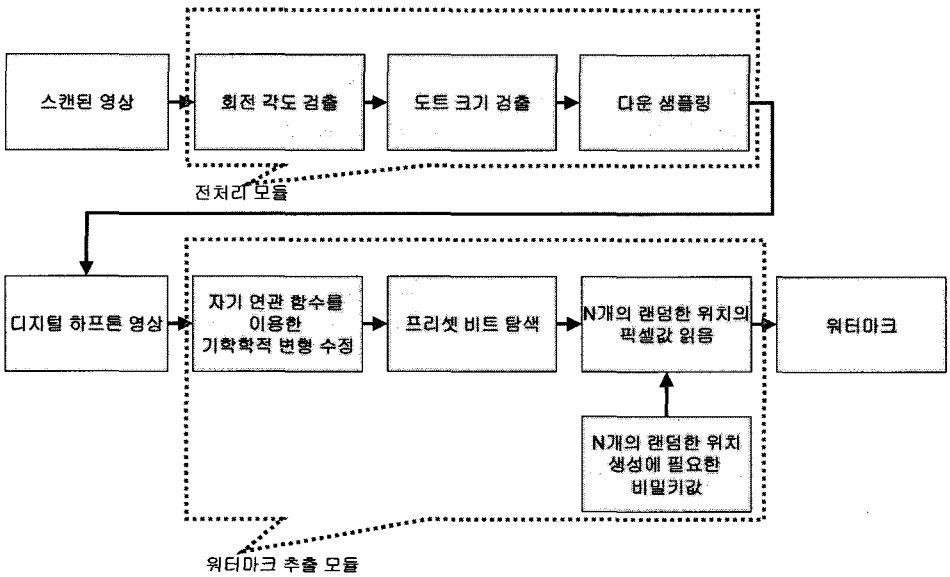


그림 6. 워터마크 추출 알고리즘

로 가정하였다. 워터마크를 담고 있는 디지털 하프톤 영상에서 직접 워터마크를 추출하는 것은 실제 인쇄, 스캔하여 워터마크를 추출하는 과정의 일부만 사용하면 쉽게 구현할 수 있다. 본 논문에서는 스캐너의 해상도를 프린터보다 더 높게 설정하는 것으로 가정하였다. 만약 스캐닝 해상도가 높지 않다면 스캐너 잡음과 스캐너의 정렬 에러 때문에 정확한 픽셀값을 복원하기가 힘들다. 따라서 인쇄된 하프톤 영상을 고해상도로 스캔하여 영상 처리 기술을 이용하여 원래의 디지털 하프톤 영상을 복원하였다.

워터마크 추출 시스템은 2가지의 모듈로 이루어진다. 첫 번째 모듈인 전처리 모듈에서는 스캔된 영상을 영상처리하여 디지털 하프톤 영상을 생성한다. 두 번째 모듈인 워터마크 추출 모듈에서는 디지털 하프톤 영상에서 워터마크를 추출한다. [그림 6]은 전체 워터마크 추출 시스템의 블록도를 보여준다.

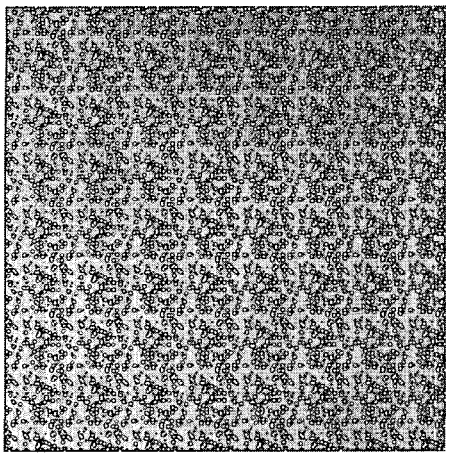
전처리 모듈에서 가장 중요한 단계는 회전 각도 검출 단계이다. 일반적으로 스캔된 영상은 회전되어 스캔될 수 있다. 따라서 이러한 회전이 있더라도 워터마크를 신뢰성 있게 추출하는 것이 매우 중요하다. 회전을 교정하는 많은 알고리즘이 있다. 본 논문에서는 하후 변환(Hough Transform)을 이용하여 회전 각도를 검출하였다. 먼저 스캔된 영상에서 간단한 에지 검출기를 이용하여 수평 에지를 검출한다. 이들 수평 에지를 이루는 픽셀들은 하후 변환기에 입력되고 하후 변환기는 이들 수평 에지의 각도를 계산해 낸다. 이 각도 정보를 이용하여 회전된 영상을 똑바르게 교정할 수 있다 [11].

각도를 교정한 다음, 워터마크 삽입에 사용하였던 비밀키를 시드키(seed key)로 하여 N개의 랜덤한 위치를 생성한다. 물론 비밀키는 워터마크 삽입자가 워터마크 검출자에게 전달하여야 한다. 하프톤 영상안에서 기본 블록이 시작되는 위치를 찾기 위하여 미리 값이 정해진 값으로 설정되어 있는 프리셋 비트(preset bits)들을 찾는다. 이러한 프리셋 비트들이 필요한 이유는 워터마크 블록이 평행이동될 수도 있기 때문이다. 즉 기본 블록의 처음 P개의 비트는 미리 알려진 값으로 설정되어 있다. 따라서 복원된 이진 영상을 처음부터 스캔하면서 N개의 랜덤한 위치 중에서 앞의 P개의 비트를 검사한다. 이 비트들이 미리 설정된 프리셋 비트 값과 같으면 기본 블록을 찾은 것이다. 미리 설정된 비트들을 찾은 후에, N개의 랜덤한 위치 중에서 P개의 프리셋 위치들을 제외한 나머지 위치들의 픽셀값들을 읽으면 바로 이것이 워터마크 비트들이 된다.

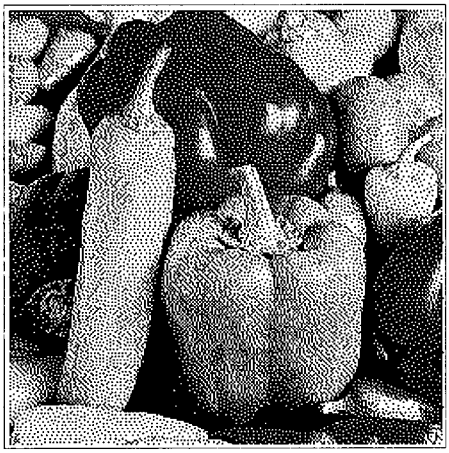
4. 실험 결과

알고리즘의 성능을 평가하기 위하여 제안된 알고리즘을 PEPPER리고 불리우는 테스트 영상에 대하여 적용하였다. 하프토닝 LUT에서는 주위 픽셀의 수 $N_1 = 15$로 하였으며 후처리 LUT에서는 $N_2 = 14$로 하였다. 따라서 하프토닝 LUT를 위해서는 $2^{N_1}2^8 = 2^{23}$ 바이트가 필요하며 후처리 LUT에 대해서는 $2^{N_2}2^8 = 2^{22}$ 바이트가 필요하다. 하프토닝을 하기 전에 각각의 LUT들을 LENNA, DEBBIE, CAMERAMAN 등의 11개의 샘플 영상을 이용하여 학습시켰다. 학습이 종료된 후에 PEPPER 영상에 대하여 전술한 알고리즘을 적용하였다. [그림 7] (a)는 PEPPER 영상을 보여준다. [그림 7] (b)는 워터마크 비트들을 보여준다. 워터마크는 서로 겹치지 않는 32×32블록들로 구성되었다. 각 블록들은 128비트의 워터마크 비트들을 가지고 있다. "0"값을 가지는 비트들은 검정색 점으로, "1"값을 가지는 비트들은 흰색 점으로 표시되어 있다. [그림 7] (c)는 워터마크를 포함하지 않고 LUT를 이용하여 하프토닝만을 한 경우의 결과 영상이다. [그림 7] (d)는 워터마크를 포함시키고 LUT를 이용하여 하프토닝을 한 경우의 결과 영상이다. [그림 7]에서 알 수 있듯이 워터마크를 포함시키더라도 결과 영상은 크게 달라지지 않는다.

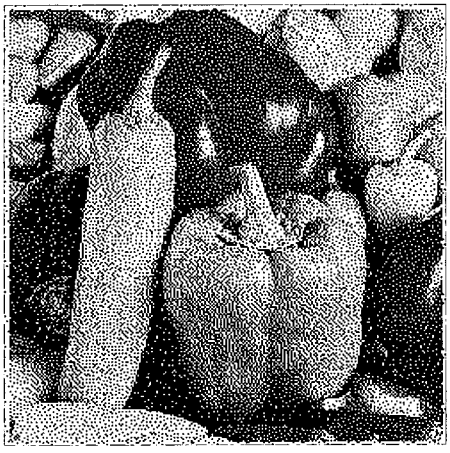
(a)



(b)



(c)



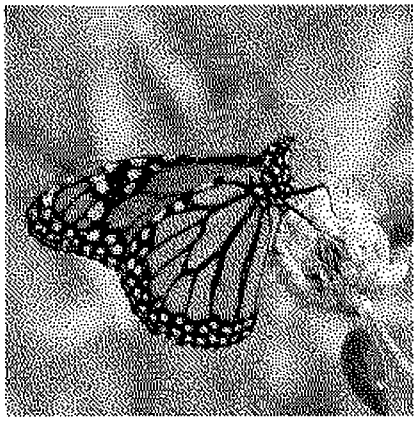
(d)

그림 7. 256×256 PEPPER 영상 워터마킹 결과. (a) 그레이 스케일 영상 (b) 워터마크 영상 (c) 워터마크를 포함하지 않은 하프토닝 결과 (d) 워터마크를 포함한 하프토닝 결과

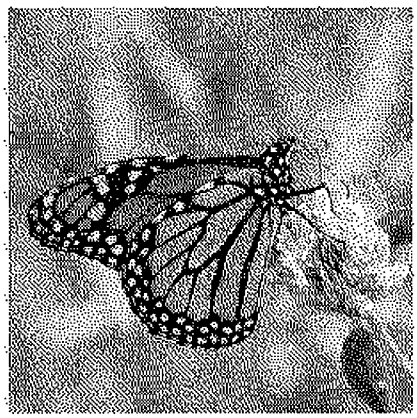
인쇄 영상 워터마킹의 다른 방법으로는 Ming등이 제안한 DHED라고 하는 방법이 있다 [6-9]. DHED는 오류확산(error diffusion) 하프토닝 과정 중에 특정한 픽셀에 강제적으로 워터마크 비트를 저장하는 워터마킹 방법이다. DHED는 하프토닝 오류를 확산하는 과정을 이용하여 강제적으로 삽입된 워터마크 비트들로 인한 왜곡을 보정하는 방법이다. DHED는 간단하고 좋은 방법이지만 많은 계산 시간을 필요로 한다. 본 논문에서는 DHED 방법과 제안된 방법을 [표 1]에서 하프토닝 품질과 하프토닝 시간을 기준으로 비교하였다. [그림 8]에서 DEBBIE 영상과 BUTTERFLY 영상에 대하여 그 결과를 비교하였다. 하프토닝 품질은 원본 영상과 하프톤 영상과의 차이를 MPSNR(Modified Peak Signal-to-Noise Ratio)로 측정하였다 [12]. [표 1]에서 알 수 있듯이 제안된 방법이 품질은 약간 떨어지지만 수행 속도가 10배 이상 빠른 장점이 있다. 또한 하프토닝 품질은 LUT를 학습시킬 때에 더 많은 학습 영상을 사용하게 되면 향상시킬 수 있다.

(a)

(b)



(c)



(d)

그림 8. DHED 방법과의 워터마킹 비교. (a) DEBBIE 영상 워터마킹(DHED) (b) DEBBIE 영상 워터마킹(제안된 방법) (c) BUTTERFLY 영상 워터마킹(DHED) (d) BUTTERFLY 영상 워터마킹(제안된 방법)

표 1. DHED 방법과 제안된 방법과의 비교

방법 \ 영상	"DEBBIE"	"BUTTERFLY"	"LENA"
DHED	23.52dB/6.89초	21.92dB/6.86초	22.16dB/6.92초
제안된 방법	23.05dB/0.47초	21.51dB/0.46초	21.50dB/0.49초
제안된 방법의 학습시간	16.02초	16.02초	16.02초

여기서 제안된 방법의 학습시간을 측정하여 보면 [표 1]과 같다. 학습시간은 약 16초로 11개의 테스트 영상과 대응되는 하프톤 영상을 이용하여 학습하는데 걸린 시간이다. 이 학습은 워터마킹을 실행하기 전에 한번만 실행하면 되고 또한 미리 학습을 시켜서 하드 디스크에 LUT의 내용을 저장하여 두고 단순히 하드 디스크에서 읽기만 할 수도 있다. 따라서 DHED가 매번 약 7초가 걸리는데 반하여 제안된 방법은 초기에 한번만 16초가 걸리고 다음부터는 매번 약 0.47초의 시간만 소요된다.

제안된 방법을 실제로 응용하는 예로 간단한 문서 위조 방지 프로그램을 작성하였다. 회사의 사원증에 들어있는 회사 로고에 사원번호를 워터마크로 삽입하고 이 로고 영상에서 사원 번호를 추출하여 인쇄된 사원번호와 대조를 할 수 있는 시스템을 실제로 스캐너와 프린터를 이용하여 구성하였다. [그림 9]에서 보듯이 스캔된 영상에서 워터마크가 정확히 추출됨을 볼 수 있다.

그림 9. 인쇄 영상 워터마킹을 응용한 신분증 위조 방지 시스템

5. 결 론

본 논문에서는 LUT를 이용한 하프톤 영상을 위한 새로운 워터마킹 알고리즘을 제안하였다. 이 알고리즘은 하프톤 영상의 랜덤한 위치에 워터마크 비트들을 저장한다. 삽입된 워터마크 비트들로 인한 하프톤 영상의 품질 저하를 막기 위하여 제안된 알고리즘에서는 LUT 기법을 사용하여 최적의 픽셀값을 찾는다. 제안된 방법을 사용하면 비교적 많은 워터마크 데이터를 삽입할 수 있으며 비교적 왜곡이 적은 인쇄영상을 유지할 수 있다.

2차원 바코드와 인쇄영상 워터마크를 비교하여 보면 2차원 바코드는 워터마크에 비하여 상대적으로 쉽게 추출이 가능한 점은 장점이지만, 생성된 그림은 의미없는 기계적인 영상이 되는데 반하여 제안된 인쇄영상 워터마크는 회사의 로고와 같은 영상 안에 정보를 저장할 수 있다는 장점을 갖는다.

참고문헌

[1] R. A. Ulichney, Digital Halftoning. MIT Press, Cambridge, MA, 1987.

[2] J. P. Allebach, "DBS: Retrospective and Future Direction," Proceedings of SPIE Vol. 4300, pp. 358-376, 2001.

[3] H. Z. Hel-Or, "Copyright Labeling of Printed Images," Proceedings of IEEE Int. Conf. on Image Processing, pp. 702-705, 2000.

[4] S.G. Wang and K.T. Knox, "Embedding digital watermarks in halftone screens," Proceedings of SPIE Vol. 3971, pp. 218-227, 2000.

[5] Z. Baharav and D. Shaked, "Watermarking of dither halftoned images," Proceedings of SPIE Vol.3657 pp. 307-316, 1999.

[6] Fu, M.S. and Au, O.C.: Data Hiding in Halftone Images by Stochastic Error Diffusion. Proceedings of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing pp. 1965-1968, 2001.

[7] Fu, M.S. and Au, O.C.: Hiding Data in Halftone Image using Modified Data Hiding Error Diffusion. Proceedings of SPIE Vol. 4067 pp.1671-1680, 2000.

[8] Fu, M.S. and Au, O.C.: Data Hiding for Halftone Images. Proceedings of SPIE Vol.3971, pp. 228-236, 2000.

[9] Fu, M.S. and Au, O.C.: Halftone image data hiding with intensity selection and connection selection. Signal Proceeding: Image Communication 16 pp.909-930, 2001.

[10] Mese, M., Vaidyanathan, P. P.: Look-Up Table (LUT) Method for Image Halftoning. Proceedings of ICIP, Vancouver, Canada, 2000.

[11] Parker, J.R.: Algorithms for Image Processing and Computer Vision. John Wiley & Sons, Hoboken, NJ, 1996.

[12] D. Kacker and J.P. Allebach, "Joint Halftoning and Watermarking," Proceedings of IEEE Int. Conf. on Image Processing, pp. 69-72, 2000.

천 인 국(In-Gook Chun) [정회원]

- 1983년 2월 : 서울대학교 전자공학과 (공학사)
- 1985년 2월 : KAIST 전기및자공학과 (공학석사)
- 1993년 2월 : KAIST 전기및전자공학과(공학박사)
- 1993년 9월 ~ 현재 : 순천향대학교 컴퓨터학부 교수

<관심분야>
영상처리, 컴퓨터비젼, 인공지능.