

반복적인 에러 최소화 기법을 이용한 하프톤 영상 워터마킹

천 인 국^{*}

요 약

본 논문에서는 반복적인 에러 최소화 기법을 이용한 하프톤 영상의 워터마킹 알고리즘을 제안하였다. 워터마크는 하프톤 영상의 랜덤한 위치의 픽셀값으로 저장된다. 삽입된 워터마크로 인한 하프톤 영상의 왜곡을 최소화하고 비가시성을 증대하기 위하여 반복적인 에러 최소화 기법이 사용된다. 원영상과 HVS(Human Visual System) 필터 처리된 하프톤 영상과의 차이를 하프톤 에러로 정의하고 반복적으로 각각의 픽셀 위치에서 이 하프톤 에러를 최소화할 수 있는 픽셀 패턴을 찾아서 이것으로 원래의 픽셀패턴을 대체한다. 절단이나 회전과 같은 기하학적인 변형에 견고하게 하기 위하여 동일한 워터마크를 반복하여 하프톤 영상 안에 삽입하였다. 실제 인쇄 및 스캐닝 실험을 통하여 제안된 알고리즘이 기하학적 변형에 견고함을 보였고 또한 기존과 방법과 비교하여 제안된 방법이 많은 양의 워터마크 정보에도 불구하고 우수한 품질의 하프톤 영상을 생성함을 보였다.

Half-tone Image Watermarking Based on Iterative Error Minimizing Method

In-Gook Chun^{*}

ABSTRACT

In this paper, we propose a new watermark algorithm for half-tone images using iterative error minimizing method. In the proposed method, watermark bits are hidden at pseudo-random locations within a half-tone image. To remove the distortions due to the inserted watermark bits and increase the invisibility of watermark, an iterative error minimizing technique is used. We define the halftoning error is defined as the difference between the original grayscale image and HVS-filtered printed half-tone image. Then we iteratively find the pixel pattern with minimum halftoning error and displace the original pixel pattern with it. In order to be robust to geometrical modification like cropping or rotation, we insert the same watermark periodically into half-tone images. Experiments using printed and scanned images show that the proposed method is a robust method to the geometrical modification and to hide the large amount of data within a half-tone image without noticeable distortion.

Key words: image watermarking(영상 워터마킹), half-tone image(하프톤 영상), DBS

1. 서 론

디지털 영상 워터마킹(digital image watermark-

※ 교신저자(Corresponding Author) : 천인국, 주소 : 충남 아산시 신창면 읍내리(336-745), 전화 : 041)530-1278, FAX: 041)530-1548, E-mail : chunik@sch.ac.kr

접수일 : 2004년 1월 5일, 완료일 : 2004년 2월 27일

^{*} 정회원, 순천향대학교 정보기술공학부 부교수

※ 본 연구과제는 2002학년도 순천향대학교 산업기술연구소 학술연구조성비 일반연구과제로 지원을 받아 수행하였음.

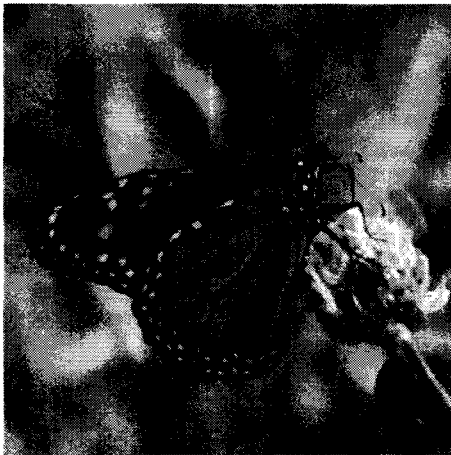
ing)은 영상 안에 저작권 정보를 보이지 않게 삽입하여 영상의 저작권을 보호하기 위한 기법이다. 영상에 대한 워터마킹 기법은 영상의 품질에 최소한으로 영향을 주면서 충분한 저작권 정보를 저장할 수 있어야 한다. 지금까지는 주로 온라인 영상, 즉 온라인상에서 파일형태로 존재하는 영상에 대한 워터마킹 연구가 주로 이루어져왔다. 그러나 인쇄에 주로 사용되는 하프톤(halftone) 영상에 대한 워터마킹 연구는

상대적으로 미진한 편이다. 현재 흑백이나 컬러를 막론하고 거의 모든 프린터에서 하프톤 기법을 이용하여 영상을 인쇄한다. 따라서 하프톤 영상에 워터마크를 삽입할 경우, 그 응용 분야는 상당히 광범위할 수 있다. 예를 들면 프린터로 인쇄된 문서를 분석하여 이를 출력한 프린터를 추적할 수도 있고, 인터넷으로 구입한 온라인 티켓을 프린터로 출력하는 경우, 티켓의 변조를 방지하는데 사용될 수 있으며, 각종 신분증의 사진에 신분정보를 넣어 신분증 위조를 방지할 수도 있다. 본 논문에서는 하프톤 영상에 대한 워터마킹 기법에 대하여 연구하였다.

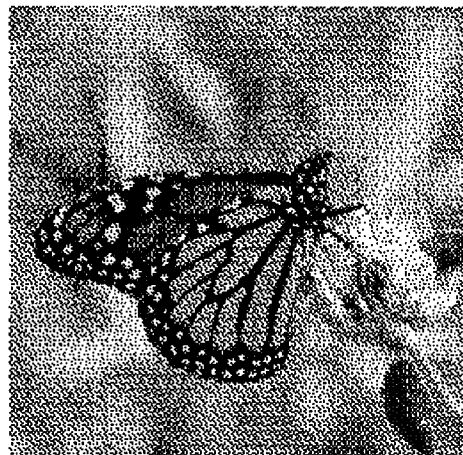
하프톤 기법은 그레이스케일 영상을 2가지의 색

상 즉 흰색과 검정색만을 이용하여 표현하는 방법이다. 현재 사용되고 있는 하프톤 기법에는 여러 가지가 있다[1,2]. 가장 많이 사용되는 하프톤 기법은 오더드 디더링(ordered dithering), 에러확산(error diffusion), 반복적 기법(iterative halftoning)등의 3가지이다. 그림 1에 여러 가지 하프톤 기법을 이용하여 생성한 하프톤 영상을 나타내었다.

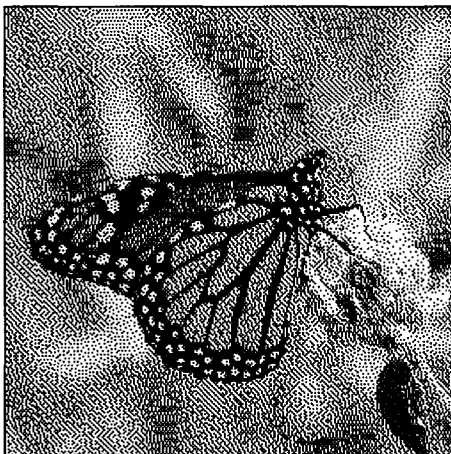
오더드 디더링은 임계값 패턴을 이용하여 영상픽셀과 임계값 패턴상의 값을 비교하여 영상픽셀값이 임계값 패턴상의 값보다 작으면 0으로 만들고 그렇지 않으면 1로 만드는 방법이다. 이 방법은 가장 간단한 방법이지만, 하프톤 영상의 품질은 그다지 좋지



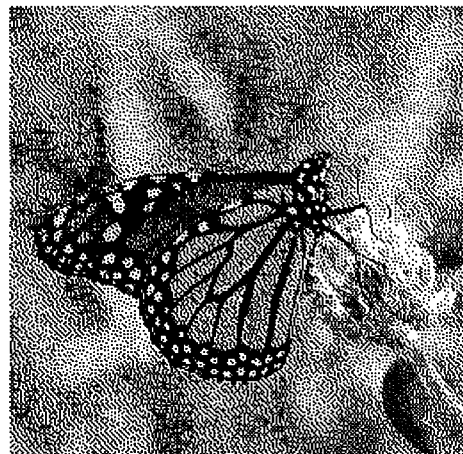
(a)



(b)



(c)



(d)

그림 1. 여러 가지 하프토닝 기법 (a) 원래의 그레이스케일 영상 (b) 오더드 디더링 하프토닝 기법 (c) 에러확산 하프토닝 기법 (d) 반복적 하프토닝 기법

않다. 에러확산은 이전의 그레이스케일 픽셀값과 하프톤 영상의 픽셀값사이의 에러를 계산하여 다음 픽셀을 하프톤할 때 이 에러를 고려하는 방식이다. 이 방식은 비교적 간단하면서 좋은 품질의 하프톤 영상을 생성하는 것으로 알려져 있다. 마지막으로 반복적인 하프톤 기법은 초기의 하프톤 영상을 생성한 다음, 이것을 여러 차례 반복하여 처리하여 최종적인 하프톤 영상을 생성하는 기법이다. 가장 잘 알려져 있는 기법은 DBS(Direct Binary Search)이다[2]. 이 방법은 많은 계산량을 요구하지만 생성되는 하프톤 영상은 품질은 매우 좋은 것으로 알려져 있다. 본 논문에서는 이 반복적인 하프톤 기법의 개념을 이용하여 워터마킹된 하프톤 영상의 왜곡을 최소화하였다.

지금까지 연구되었던 워터마킹 기법들은 주로 그레이스케일 영상 또는 컬러영상에 대하여 제안된 방법이기 때문에 검정과 흰색 픽셀만을 가지고 있는 하프톤 영상에 대해서는 직접 적용할 수 없다. 지금까지의 하프톤 영상의 워터마킹 연구는 크게 2가지로 나누어진다. 첫번째 방법은 오더드 디더링을 이용하여 하프톤에서 1개가 아닌 2개의 임계값 패턴을 사용하는 방식이다. 즉 워터마크 비트가 0이나 1이나에 따라 사용되는 임계값 패턴을 다르게 하는 방법이다[3-5]. 이 방법은 구현하기가 비교적 간단하지만 어떻게 신뢰성있게 워터마크 비트를 추출할 것인지와 어떻게 최적의 임계값 패턴 쌍을 결정할 것인지가 문제이다. 두번째 방법은 하프톤 영상이 이진 영상이므로 워터마크 비트를 직접 하프톤 영상의 일부 픽셀값으로 저장하는 방법이다. Ming과 Oscar는 주로 이 방식을 이용하여 에러확산을 이용하는 하프톤 기법에 대한 여러 가지 워터마킹 기법을 발표하였다[6-9]. 그들이 발표한 방법 중 가장 대표적인 것이 DHED(Data Hiding Error Diffusion)로서 이는 워터마킹과 에러확산을 결합시킨 것이다. 즉 하프톤 영상의 일부 픽셀을 워터마크 정보에 따라 강제적으로 0 또는 1로 만들고 이로 인한 하프톤 에러를 다음 하프톤 픽셀을 생성하는데 참조하는 방식이다. 이 방식은 간단하다는 장점이 있지만 하프톤 에러가 스캔 순서에 따른 다음 픽셀들에 의해서만 에러가 보정되는 약점을 가진다. 따라서 하프톤 에러가 전체 이웃 픽셀에 의해서 보정이 되지 않기 때문에 왜곡이 상당부분 남아있게 된다.

본 논문에서는 하프톤 에러 최소화 기법에 의한

하프톤 영상의 워터마킹 기법을 제안하였다. 여기서는 영상의 품질에 최소한도의 영향만을 주기 위하여 반복적인 하프톤 에러 최소화 기법을 사용한다. 반복적인 하프톤 에러 최소화 기법이란 반복적인 이진 패턴 탐색을 통하여 최적의 이웃 픽셀을 찾아서 하프톤 에러를 최소화하는 방법이다. 이 방법의 최대 약점은 계산시간이 많이 걸린다는 점이다. 그러나 실험을 통하여 대개 10번 정도의 반복이 지나면 결과가 빠르게 수렴함을 알 수 있었다. 또한 의도적이지 않은 기하학적인 변형에 강인하게 만들기 위하여 동일한 워터마크가 반복하여 삽입되는 워터마크 패턴을 제작하였다. 스캔된 하프톤 영상에서의 워터마크의 추출은 2단계로 이루어져 있다. 첫번째 단계는 스캔된 하프톤 영상에서 여러 단계의 영상처리 과정을 통하여 이진 하프톤 영상을 얻는 과정이다. 두번째 단계는 이진 하프톤 영상에서 워터마크 비트들을 추출하는 단계이다. 만약 스캔시의 해상도가 프린트시의 해상도보다 높다면 워터마크는 신뢰성있게 검출될 수 있다. 제안된 알고리즘은 완전한 블라인드 워터마킹 알고리즘으로서 난수 발생기의 시드값 이외에는 아무런 다른 정보를 요구하지 않는다. 본 논문에서는 실제로 프린터와 스캐너를 이용하여 인쇄, 스캔된 영상에 대하여 실험을 하고 그 결과를 제시하였다.

2. 하프톤 영상 워터마킹 알고리즘

2.1 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. 즉 그림 2와 같이 워터마크 비트가 0이나 1이나에 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.

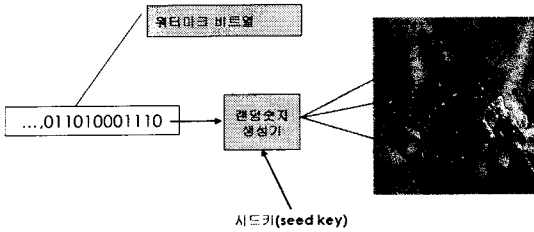


그림 2. 워터마크의 삽입: 워터마크는 시드값에 의해 생성되는 랜덤한 위치의 픽셀값으로 저장된다.

여기서의 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 알고리즘에서는 반복적인 탐색에 의하여 왜곡을 줄이는 최적의 이웃 픽셀 패턴을 선택함으로써 이러한 왜곡을 최소화한다. 본 논문에서 제안한 하프톤 영상의 워터마크 삽입 알고리즘의 전체 블록도는 그림 3과 같다.

워터마크 삽입과정은 크게 나누어 초기 하프토닝 단계, 워터마크 삽입 단계, 에러 최소화 단계의 3개의 단계로 이루어져 있다. 초기 하프토닝 단계에서는 오더드 디터링 방법을 이용하여 원래의 그레이스케일 영상으로부터 주어진 임계값 패턴을 사용하여 초기 하프톤 영상을 만든다. 이 초기 하프톤 영상은 뒤에

에러 최소화 단계를 거치면서 점점 더 나은 하프톤 영상으로 변화된다.

두번째 단계인 워터마크 삽입 단계에서는 삽입자와 추출자만이 알고 있는 비밀키를 이용하여 N개의 랜덤한 위치를 생성한다. 워터마크 데이터는 바로 이 N개의 랜덤한 위치에 저장된다. 본 논문에서 워터마크는 이진 비트열로 가정한다. N개의 랜덤한 위치들의 픽셀값을 워터마크 비트의 값에 따라 0 또는 1로 강제적으로 만들어서 워터마크 데이터를 저장한다.

에러 최소화 단계에서는 이전 단계에서 생성된 초기 하프톤 영상의 화질을 반복적인 절차를 통하여 개선시킨다. 워터마크 픽셀을 제외한 나머지 픽셀 값은 원영상과 하프톤 영상간의 하프톤 에러를 줄이는 방향으로 변경된다. 즉 워터마크가 들어있지 않은 픽셀들의 값을 여러 가지로 바꾸어 본 다음, 그중에서 가장 에러가 최소인 픽셀 패턴을 선택한다. 하프톤 에러는 다음과 같이 정의된다[10,11].

$$E = \sum_m \sum_n |h(m, n) * g(m, n) - h(m, n) * f(m, n)|^2$$

위의 식에서 위의 식에서 $h(m, n)$ 은 HVS(Human Visual System) 필터의 PSF(Point Spread Function)를 나타내고 $g(m, n)$ 은 픽셀의 값이 0이나 1인 하프톤 영상을, *은 2차원 컨볼루션을 나타낸다. 또한 $f(m, n)$ 은 픽셀의 값이 0.0에서 1.0사이에 있는 원래의 그레이스케일 영상을 나타낸다. 위의 식에서 $h(m, n) * g(m, n)$ 은 인간의 눈으로 본 하프톤 영상을 나타내고, $h(m, n) * f(m, n)$ 은 인간의 눈으로 본 원래의 그레이스케일 영상을 나타내며, $|h(m, n) * g(m, n) - h(m, n) * f(m, n)|^2$ 는 인간의 눈으로 본 하프톤 영상과 원래의 그레이스케일 영상 간의 에러를 나타낸다.

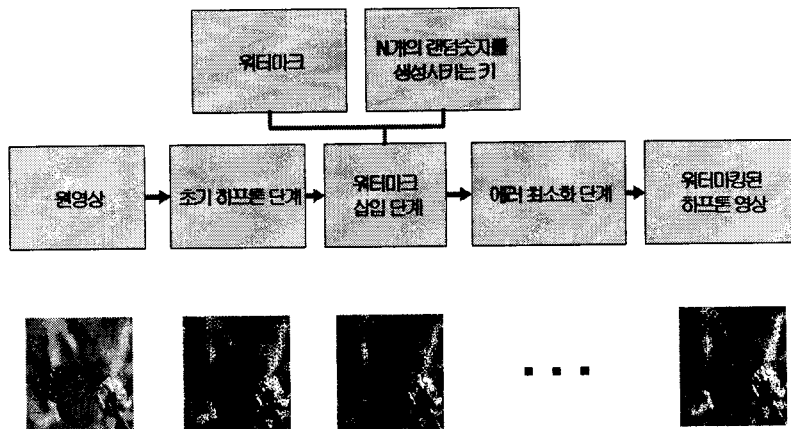


그림 3. 하프톤 영상 워터마크 삽입 알고리즘의 블록도

$g(m, n) - h(m, n) * f(m, n)$ 은 인간의 눈으로 본 하프톤 영상과 원래의 그레이스케일 영상간의 차이 즉 하프톤 영상의 에러를 나타낸다. 이 에러가 감소될수록 인간은 하프톤 영상이 원래의 그레이스케일 영상과 비슷하다고 여긴다. 따라서 하프톤 에러가 감소할수록 품질이 좋은 하프톤 영상이라고 말할 수 있다.

다음 문제는 어떤 기법을 사용하여 위의 에러를 최소로 줄이는 하프톤 영상을 찾느냐 하는 것이다. 에러를 최소화하는 절차는 DBS(Direct Binary Search)기법을 사용하였다 [10]. DBS는 픽셀의 값을 역전(toggle)시키거나 픽셀의 값을 서로 교환(swap)하여 여러 개의 후보 픽셀 패턴을 만든 다음, 이 후보 패턴 중에서 에러를 최소로 만드는 픽셀 패턴을 선택한다. 전체 하프톤 영상은 왼쪽 상단에서 오른쪽 하단까지 순차적으로 스캔되고 각각의 픽셀 위치에서

가능한 후보 패턴들이 역전이나 교환에 의하여 생성된다. 이들 후보패턴들의 하프톤 에러가 계산되고 이 중에서 가장 에러가 적은 패턴이 현재의 패턴을 대체하게 된다. 이러한 과정이 하프톤 영상이 수렴할 때까지 반복된다. 만약 하나의 반복과정에서 한번도 픽셀 패턴의 변화가 없으면 알고리즘이 수렴한 것으로 판단한다. DBS는 근본적으로는 지역 최소값(local minimum)으로 수렴하지만 실험에서 초기값을 다르게 하여도 최종의 결과에는 별 영향을 끼치지 않는 것으로 관찰되었다. 다음의 코드는 이 에러 최소화 과정을 의사코드로 정리한 것이다.

그림 5는 이러한 에러 최소화 과정을 그림으로 설명한 것이다. 그림 5에서는 특정한 픽셀 위치에서 4개의 후보 패턴이 생성되고 이 중에서 가장 에러 값이 작은 4번째 패턴이 선택되어 현재의 패턴을 대체함을 보여주고 있다.

워터마크 추출시 발생할 수 있는 기하학적인 변형에 견고하게 만들기 위하여 주기적으로 반복되는 워터마크를 구성하였다. 다음 그림은 워터마크가 여러 개의 기본 워터마크 블록들로 이루어져 있음을 보여준다. 또한 기본 워터마크는 미리 값들이 정해져있는 프리셋 비트(preset bit)를 포함하고 있다. 이 프리셋 비트들은 워터마크 추출시 기본 워터마크 블록을 탐색하는데 사용된다.

반복적인 워터마크 데이터는 워터마크 추출시에 여러 가지로 이용된다. 먼저 워터마크 데이터가 반복적으로 들어있으면 어느 한 부분이 절단되어 사라진

```

generate an initial halftone image;
while(any pixel is changed) {
  for all pixel location (x,y) in halftone image {
    if (x,y) is one of pseudo-random locations
      continue;
    else {
      for all toggled and swapped patterns at (x,y)
        calculates the halftone error of that pattern;
      choose the minimum error pattern;
      replace the block by the minimum error pattern;
    }
  }
}
    
```

그림 4. 에러 최소화 과정의 의사코드 알고리즘

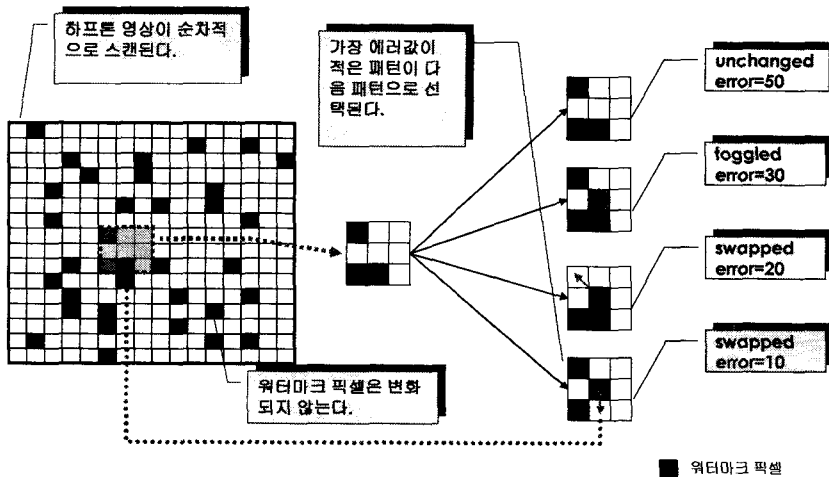
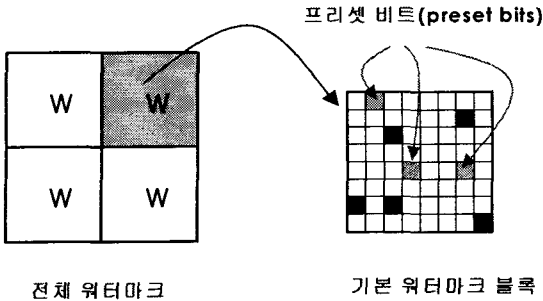


그림 5. 에러 최소화 과정



전체 워터마크 기본 워터마크 블록
 그림 6. 워터마크의 구조

다고 하더라도 다른 부분을 이용하여 워터마크를 신뢰성있게 추출할 수 있다. 또한 워터마크를 반복적으로 구성하면 이것을 이용하여 기본 워터마크 블록의 크기를 계산할 수 있다. 이는 워터마크가 포함된 하프톤 영상의 자기연관함수(autocorrelation function)를 구하면 알 수 있다 [12]. 즉 하프톤 영상안에 반복되는 데이터가 들어 있으면 자기연관함수는 반복되는 횟수에 비례하는 여러 개의 피크값을 가진다. 이들 피크값 간의 거리가 바로 기본 워터마크 블록의 크기가 된다. 또한 추가적으로 하프톤 영상에 기하학적인 변형이 가해졌을 경우 이 기하학적 변형의 파라미터들을 피크값들을 이용하여 추출할 수 있다. 즉 그림 7처럼 하프톤 영상이 기하학적으로 변형되면 이들 피크값들도 따라서 변형되고 따라서 피크값들의 배치를 이용하여 기하학적 변형의 파라미터를 계산할 수 있다.

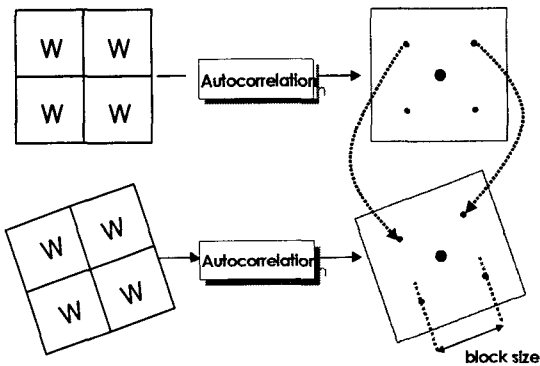


그림 7. 자기연관함수를 이용한 기하학적 변형 파라미터 계산

2.2 워터마크 추출

워터마크의 추출은 워터마크를 담고 있는 하프톤 영상을 프린터로 인쇄한 다음, 이 인쇄된 하프톤 영

상을 다시 스캐너로 스캔하여 여기에서 워터마크를 추출하는 것으로 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 알고리즘은 2가지의 큰 모듈로 나누어진다. 첫 번째 모듈은 스캔된 영상을 영상 처리하여 디지털 하프톤 영상을 생성하는 전처리 모듈이다. 두 번째 모듈은 디지털 하프톤 영상에서 워터마크를 추출하는 워터마크 추출 모듈이다. 그림 8은 전체 워터마크 추출 시스템의 블록도를 보여준다.

전처리 모듈의 첫 번째 단계는 회전 각도 검출 단계이다. 일반적으로 스캔된 영상은 회전되어 스캔될 수 있다. 따라서 이러한 회전이 있더라도 워터마크를 신뢰성있게 추출하는 것이 매우 중요하다. 회전을 교정하는 많은 알고리즘이 있다 [13]. 본 논문에서는 하후 변환(Hough Transform)을 이용하여 회전 각도를 검출하였다. 먼저 스캔된 영상에서 간단한 에지 검출기를 이용하여 수평 에지를 검출한다. 수평 에지 픽셀들은 하후 변환기에 입력되고 하후 변환기는 이들 수평 에지의 각도를 계산해 낸다. 이 각도 정보를 이용하여 회전된 영상을 똑바르게 할 수 있다. 그림 9에서 회전의 각도를 구하는 과정을 보여주고 있다.

그 다음 단계는 하프톤에 사용된 도트(dot)의 크기를 검출하는 단계이다. 스캔시의 해상도가 인쇄시의 해상도보다 크기 때문에 스캔된 영상에서의 도트는 하나의 픽셀이 아니고 여러 개의 픽셀로 되어 있다. 본 논문에서는 하프톤에 사용된 도트의 형상은 직사각형이라고 가정하였다. 실제로는 많은 다른 형상의 도트가 사용되고 있지만 간결성을 위하여 본 논문에서는 도트가 직사각형이라고 가정한다. 도트의 크기를 찾기 위하여 먼저 연결성분(connected component)을 찾아서 각 연결성분의 크기를 기록한다. 일반적으로 하프톤 영상에서는 하나의 도트로만 이루어진 많은 연결성분이 있음을 가정할 수 있기

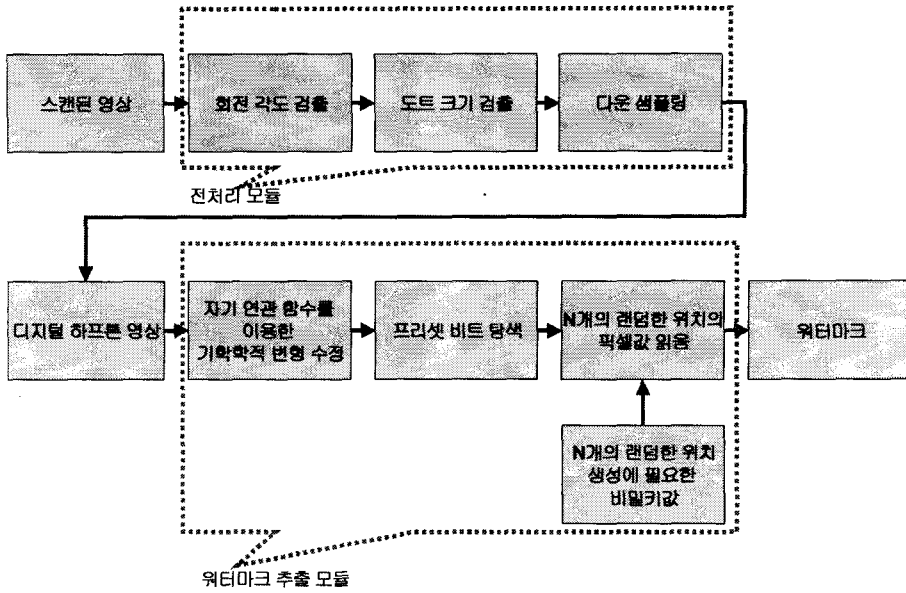


그림 8. 하프톤 영상 워터마크 추출 알고리즘의 블록도

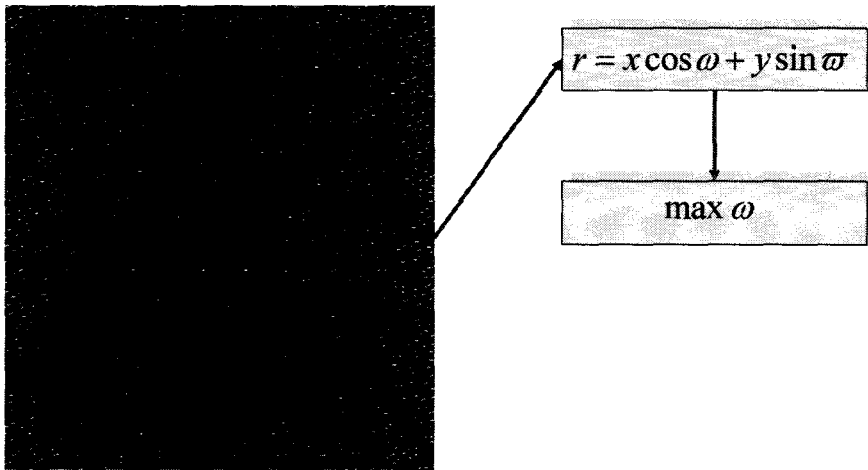


그림 9. 스캔된 하프톤 영상의 회전 각도 검출

때문에 연결성분의 크기 중에서 가장 빈도가 많은 것을 찾으면 바로 이것이 도트의 크기가 된다. 그림 10에서 하프톤에 사용된 도트의 크기를 구하는 과정을 보여주고 있다.

이런 식으로 도트의 크기를 알아낸 다음에는 에지 정보를 이용하여 도트의 중심을 알아낸다. 먼저 모든 픽셀위치에서 이웃한 픽셀간의 차이를 이용하여 x방향 에지값 $e_x(x, y)$ 과 y방향 에지값 $e_y(x, y)$ 을 구한다. 다음으로 이들 값을 수직과 수평방향으로 투상한

값들인 $e_{mx}(x)$ 과 $e_{my}(y)$ 를 구하고 이 합들이 지역적으로 최대가 되는 지점들을 구하면 이 지점이 각각의 도트들의 경계가 된다.

$$e_{mx}(x) = \sum_y e_x(x, y)$$

$$e_{my}(y) = \sum_x e_y(x, y)$$

구해진 도트 경계들의 중심점을 찾으면 그곳이 바로 각 도트들의 중심점이 된다. 다음으로 도트 중심점에 있는 픽셀값을 읽으면 그 픽셀값이 그 도트가

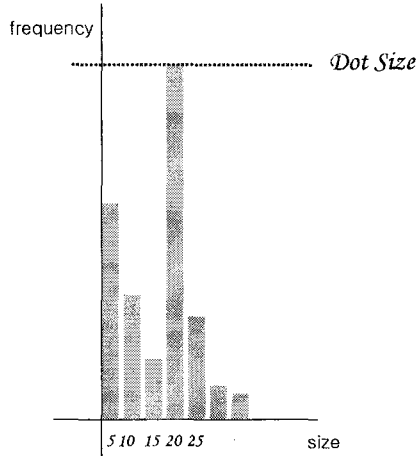


그림 10. 스캔된 하프톤 영상에서 하프톤 도트 크기 검출

흰색 도트인지 검정색 도트인지를 알려준다. 따라서 이런 식으로 스캔된 하프톤 영상을 처리하면 인쇄되기 전의 원래의 디지털 하프톤 영상을 복원할 수 있다. 그림 11에서 검출된 각 도트들의 중심점이 작은 점으로 표시되어 있다.

위터마크 추출 모듈은 복원된 디지털 하프톤 영상에서 위터마크를 추출한다. 먼저 기본 위터마크 블록의 크기를 알기 위하여 다음과 같은 자기연관 함수 (autocorrelation function)를 계산한다.

$$R_{g,g}(u, v) = \sum_m \sum_n g(m, n)g(m+u, n+v)$$

여기서 $g(m, n)$ 은 위터마킹된 하프톤 영상을 나타낸다. 만약 위터마킹된 영상 안에 주기적인 데이터

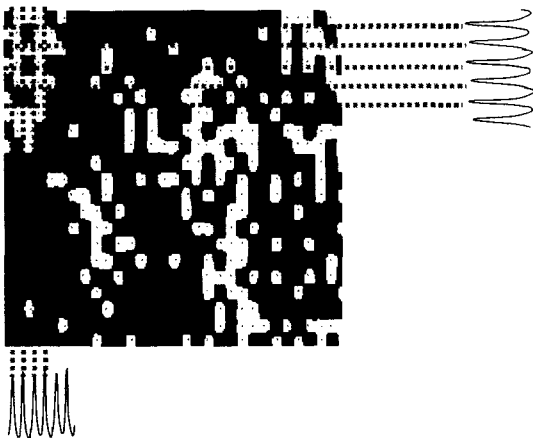


그림 11. 스캔된 하프톤 영상에서 하프톤 도트 경계와 중심점 검출

가 존재한다면 자기연관 함수값은 여러 개의 피크값을 가지게 되고 따라서 이들 피크값을 이용하여 주기적인 데이터 블록의 크기를 계산할 수 있다. 즉 피크값사이의 간격이 주기적인 데이터 블록의 크기가 된다. 피크값들은 그라디언트(gradient)를 계산하여 알 수 있다. 그림 12가 이 개념을 설명하고 있다. 이 자기연관 함수를 이용하면 추가적으로 기하학적 변형에 대하여 수정을 할 수 있다.

기본 블록의 크기를 검출한 다음, 위터마크 삽입에 사용하였던 비밀키를 시드키(seed key)로 하여 N개의 랜덤한 위치를 생성한다. 물론 비밀키는 위터마크 삽입자가 위터마크 검출자에게 전달하여야 한다. 하프톤 영상안에서 기본 블록의 위치를 찾기 위하여 프리셋 비트(preset bits)들을 찾는다. 이러한 프리셋 비트들이 필요한 이유는 위터마크 블록이 평행이동될 수도 있기 때문이다. 즉 기본 블록의 처음 P개의 비트는 미리 알려진 값으로 설정되어 있다. 따라서 복원된 이진 영상을 처음부터 스캔하면서 랜덤한 위치 중에서 앞의 P개의 비트를 검사한다. 이 비트들이 미리 설정된 비트 값과 같으면 기본 블

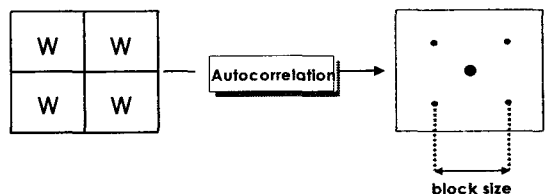


그림 12. 자기연관 함수를 이용한 기본 위터마크 블록의 크기 검출

록을 찾은 것이다. 복원된 디지털 하프톤 영상에서 기본 블록을 찾을 때까지 이 과정을 되풀이 한다. 미리 설정된 비트들을 찾은 후에, N개의 랜덤한 위치 중에서 P개의 프리셋 위치들을 제외한 나머지 위치들의 픽셀값들을 읽으면 바로 이것이 워터마크 비트들이 된다.

3. 실험 결과 및 고찰

제안한 하프톤 워터마킹 알고리즘의 성능을 평가하기 위하여 컴퓨터로 알고리즘을 구현하여 실험하였다. 실험결과는 Ming 등이 제안한 DHED(Data Hiding Error Diffusion) 알고리즘과 비교되었다. DHED는 본 논문에서 제안된 방법과 비슷하게 하프톤 영상의 픽셀에 워터마크 비트를 직접 저장하는 방식으로 다른 점은 에러 최소화 과정이 아니라 에러 확산(error diffusion) 하프톤 방법을 사용한 점이다.

첫 번째 테스트에 사용된 영상은 그림 13(a)에 보여지는 "constant"라고 불리는 128x128크기의 8비트 그레이스케일 영상이다. 이 영상은 일정한 픽셀값으

로만 이루어져 있다. 그림 13(b)에서 보듯이 워터마크는 4개의 기본 블록으로 이루어져 있으며 각각의 기본 블록은 16비트의 워터마크 비트를 가지고 있다. 그림에서 워터마크 비트 "0"는 검정점으로 표시되어 있고 비트 "1"은 흰색점으로 표시되어 있다. 그림 13(c)는 오더드 디더링에 의하여 생성된 초기 하프톤 영상을 보여준다. "salt and pepper"스타일의 잡음을 볼 수 있으며 또한 검정점들과 흰색점들의 분포가 균일하지 않다. 이것들은 주로 삽입된 워터마크 비트들로 인하여 발생한다. 그림 13(d)는 제안된 알고리즘에 의하여 생성된 최종 하프톤 영상을 보여준다. 그림에서 보듯이 검정점들과 흰색점들의 분포가 초기 하프톤 영상에서 비하여 더욱 균일함을 알 수 있다. 그리고 "salt and pepper"스타일의 왜곡도 훨씬 줄어들어 있음을 알 수 있다. 더욱더 중요한 것은 워터마크 픽셀을 비워터마크 픽셀과 구별하는 것이 거의 불가능해졌다는 것이다. 그림 13(e)는 워터마크 데이터를 갖지 않는 최종 하프톤 영상이다. 그림 13(d)와 그림 13(e)를 비교해보면 큰 차이점은 없음을 알 수 있다. 따라서 제안된 알고리즘은 워터마크

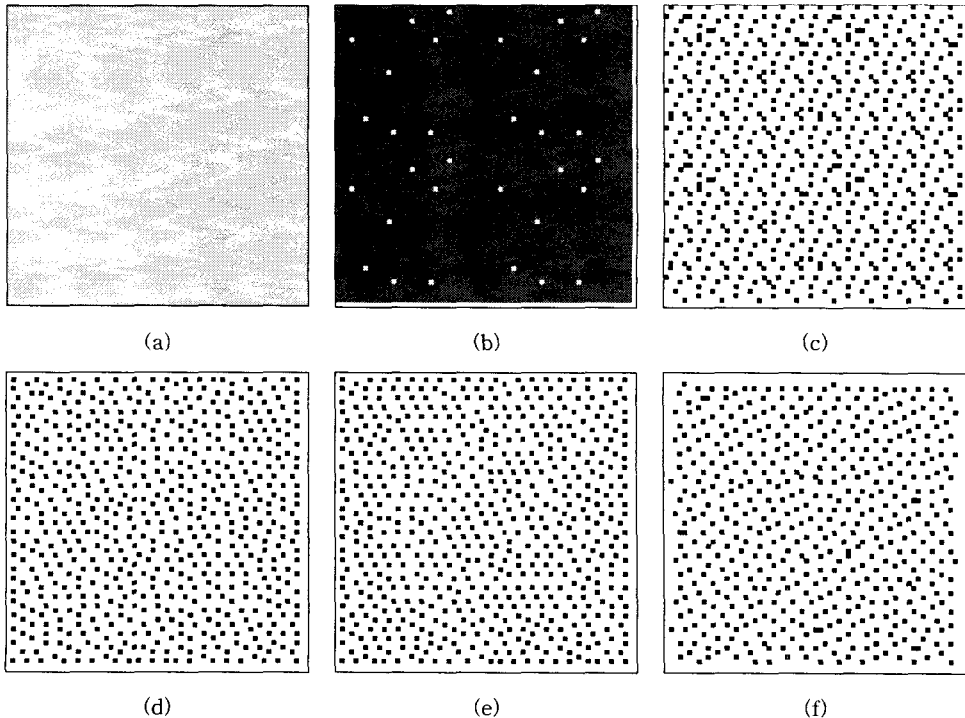


그림 13. "constant" 영상의 워터마킹 결과 (a) 원영상 (b) 워터마크 영상 (c) 초기 하프톤 영상 (d) 최종 하프톤 영상 (e) 워터마크가 없는 최종 하프톤 영상 (f) DHED를 사용한 최종 하프톤 영상

비트를 효과적으로 숨긴다는 것을 알 수 있다. 그림 13(f)는 DHED 알고리즘의 결과이다. 그림에서 보듯이 제안된 알고리즘이 DHED보다 영상의 품질 면에서 더 우수함을 알 수 있다.

다음 영상은 그림 14(a)에 보여지는 128×128의 8비트 그레이스케일 영상 “butterfly”이다. 그림 14(b)는 워터마크 영상으로 4개의 기본 블록으로 이루어져 있고 각 기본 블록은 동일한 96비트의 워터마크를

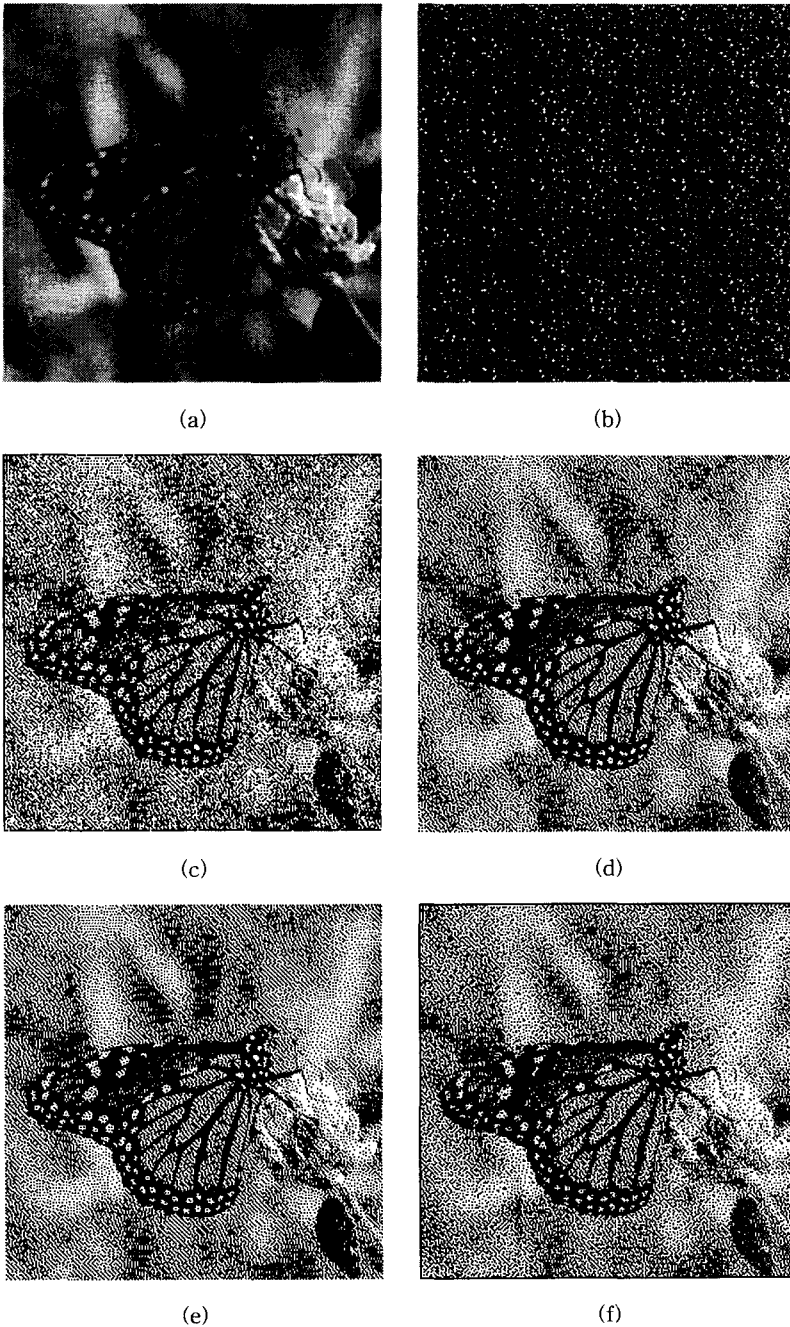


그림 14. “butterfly” 영상의 워터마킹 결과 (a) 원영상 (b) 워터마크 영상 (c) 초기 하프톤 영상 (d) 최종 하프톤 영상 (e) 워터마크가 없는 최종 하프톤 영상 (f) DHED를 사용한 최종 하프톤 영상

가지고 있다. 그림 14(c)는 오더드 디더링에 의한 초기 하프톤 영상을 보여준다. 그림 14(d)는 제안된 알고리즘으로 생성된 최종 하프톤 영상이다. 그림에서 보듯이 검정점과 흰점이 초기 하프톤 영상보다 훨씬 균일하고 "salt and pepper" 타입의 잡음이 상당히 줄어들었음을 볼 수 있다. 그림 14(e)는 워터마크를 갖지 않은 최종 하프톤 영상이다. 그림 14(d)와 (e)는 별 차이가 없음을 알 수 있다. 그림 14(f)는 DHED 알고리즘의 최종 결과이다. 그림 14(d)와 비교해보면 제안된 알고리즘이 더 우수함을 알 수 있다.

제안된 알고리즘이 절단, 회전, 평행이동과 같은 기하학적인 공격에 대하여 견고하다는 것을 테스트 하기 위하여 다음과 같은 실험을 하였다. 먼저 HP LaserJet 프린터를 이용하여 600dpi로 워터마크가 포함된 최종 하프톤 영상을 인쇄하였다. 이 인쇄된 하프톤 영상을 다시 HP scanjet을 이용하여 2400dpi로 스캔하여 그림 15(a)와 같은 영상을 얻었다. 스캐닝하기 전에 고의적으로 인쇄 영상을 회전시켰고 또한 일부를 제거하였다. 이 스캔된 영상에서 수평 에지를 구하고 이 수평 에지를 대하여 하후 변환을 하

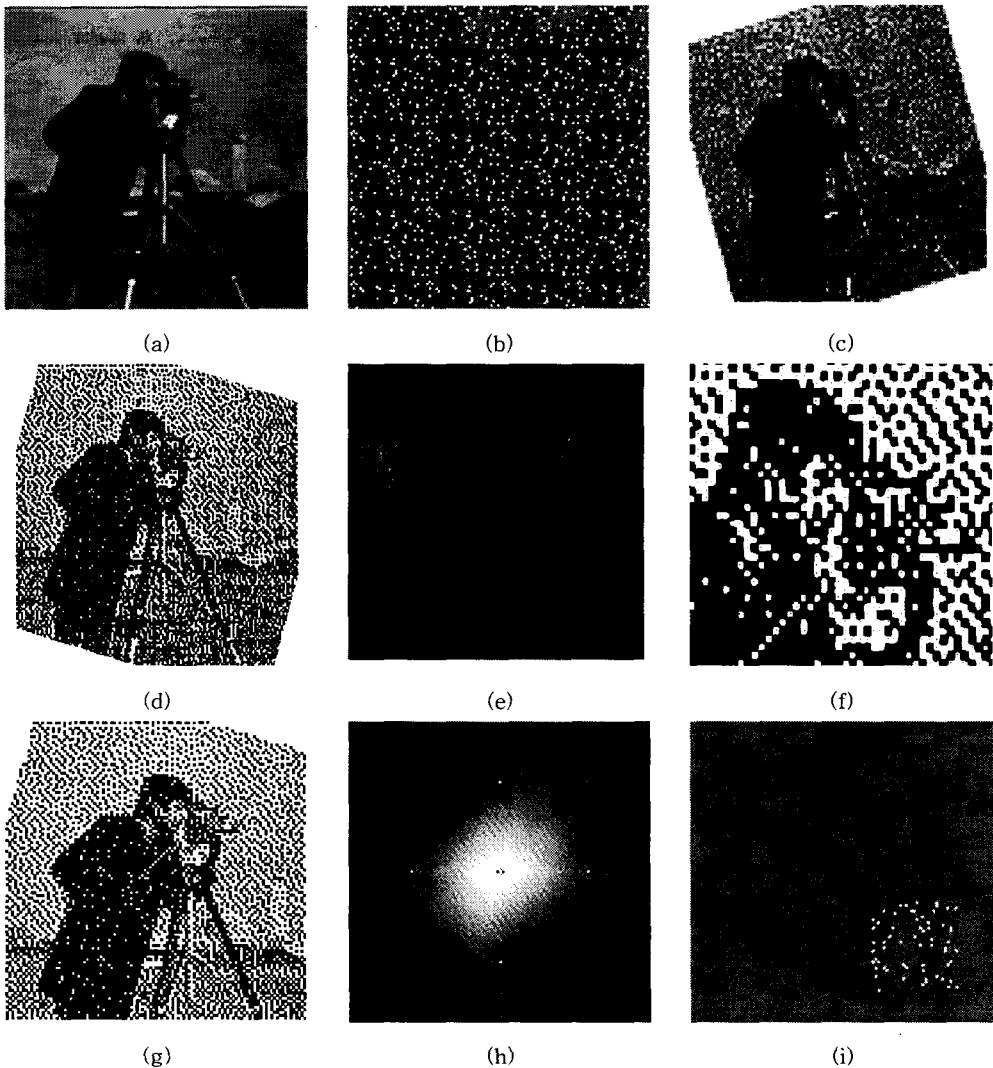


그림 15. "cameraman" 영상의 워터마크 추출 과정 (a) 원영상 (b) 삽입된 워터마크 (c) 스캔된 영상 (d) 회전 보정 영상 (e) 연결 성분 검출 영상 (f) 하프톤 도트의 중심점 검출 (g) 복원된 디지털 하프톤 영상 (h) 자기 연관 함수값 영상 (i) 추출된 워터마크 영상

표 1. 실험 결과 하프톤 영상들의 MPSNR 값 비교

	"constant" 영상	"butterfly" 영상	"cameraman" 영상
DHED	17.74dB	20.81dB	19.00dB
제안된 방법	23.63dB	22.17dB	20.72dB

여 회전된 각도를 검출하고 회전된 각도만큼 역회전 변환을 하여 그림 15(b)를 얻었다. 다시 이 영상에 대하여 연결성분을 찾은 것이 그림 15(c)이다. 연결성분을 이용하여 하프톤에 사용된 도트의 크기를 알아낸 다음, 에지 값들의 수평, 수직으로의 투사된 값들을 이용하여 각 도트들의 중심값을 알아낼 수 있다. 그림 15(d)에 도트들의 중심이 작은 검정점으로 표시되어 있다. 다음으로 각 도트들의 흰색점인지 검정색 점인지를 알아내야 한다. 그림 15(e)에서 각 도트들의 중심값을 읽어서 최종적으로 디지털 하프톤 영상을 복원한다. 다음 단계로 워터마크를 검출하기 위하여 이 이진 하프톤 영상의 자기연관함수를 계산한다. 그림 15(f)에서 보듯이 반복적인 데이터가 있으면 자기연관함수는 반복되는 회수만큼의 피크값을 가지게 된다. 이 피크값들의 거리를 이용하여 하나의 워터마크 블록의 크기를 계산할 수 있다. 워터마크 블록의 크기가 계산되면 디지털 하프톤 영상을 처음부터 스캔하여 워터마크 블록에서의 미리 설정된 비트들을 찾는다. 이 미리 설정된 비트가 발견되면 그 다음 랜덤 위치의 픽셀값을 읽으면 그것이 바로 워터마크 비트들이 된다.

논문에서 제안된 방법과 DHED 알고리즘의 결과를 수치적으로 비교하기 위하여 MPSNR(Modified Peak Signal-to-Noise Ratio)을 사용하였다. MPSNR은 원래의 그레이스케일 영상과 하프톤 영상과의 유사함을 측정하는 방법으로서 M.S.Fu 등이 제안하였다 [8]. 이는 하프톤 영상에 로우패스 필터링을 하여 그레이스케일 영상으로 변경하여 원래의 그레이스케일 영상과의 PSNR을 측정하는 것이다. 표 1은 실험영상에 대한 MPSNR을 보여준다. 표 1에서 보듯이 모든 실험영상에서 제안된 알고리즘이 DHED보다 우수하다.

4. 결 론

본 논문에서는 반복적인 하프톤 에러 최소화 기법을 이용한 하프톤 영상을 위한 새로운 워터마킹 알고리즘을 제안하였다. 이 알고리즘은 하프톤 영상의 랜

덤한 위치에 워터마크 비트들을 저장한다. 삽입된 워터마크 비트들로 인한 하프톤 영상의 품질 저하를 막기 위하여 제안한 알고리즘에서는 반복적인 하프톤 에러 최소화 기법을 이용하여 하프톤 영상의 픽셀값을 변화시켜서 최소 하프톤 에러를 만드는 픽셀값들의 배치를 찾는다. 또한 절단이나 회전과 같은 기하학적 변형에 대한 견고성을 가지게 하기 위하여 동일한 워터마크 비트 블록을 하프톤 영상에 반복적으로 삽입하였다. 이런 식으로 하프톤 영상에 워터마크를 삽입함으로써 임계값 패턴을 변화시키는 방법보다 더 많은 워터마크 데이터가 삽입될 수 있고 그림에도 불구하고 비교적 하프톤 영상의 품질이 우수하게 유지된다.

본 논문에서 제안한 알고리즘의 성능을 평가하기 위하여 몇 개의 테스트 영상을 이용하여 컴퓨터 모의 실험을 수행하였다. 하프톤 영상에 워터마크를 삽입하여 이를 실제로 프린터를 이용하여 인쇄하고 이를 다시 스캐너를 이용하여 스캔한 다음, 이 스캔된 영상으로부터 워터마크를 성공적으로 추출하였다. 이 실험에서 제안된 워터마킹 알고리즘이 기존의 알고리즘인 DHED에 비하여 워터마크 비가시성 및 견고성 면에서 우수함을 확인할 수 있었다. 제안된 워터마킹 알고리즘은 지폐나 쿠폰, 온라인 티켓, 신분 식별 카드 등의 보안이 필요한 문서를 인쇄할 때 사용될 수 있다.

참 고 문 헌

- [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," *Pro-*

- ceedings 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] M.S. Fu and O.C. Au, "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] M.S. Fu and O.C. Au, "Hiding Data in Halftone Image using Modified Data Hiding Error Diffusion," *Proceedings of SPIE Vol. 4067*, pp.1671-1680, 2000.
- [8] M.S. Fu and O.C. Au, "Data Hiding for Halftone Images," *Proceedings of SPIE Vol.3971*, pp. 228-236, 2000.
- [9] M.S. Fu and O.C. Au, "Halftone image data hiding with intensity selection and connection selection". *Signal Proceeding: Image Communication 16*, pp. 909-930, 2001.
- [10] D. Kacker and J.P. Allebach, "Joint Halftoning and Watermarking," *Proceedings of IEEE Int. Conf. on Image Processing*, pp. 69-72, 2000.
- [11] M. Mese, and P.P Vaidyanathan, "Look-Up Table (LUT) Method for Inverse Halftoning," *IEEE Trans.on Image Processing, Vol. 10*, pp. 1566-1578, 2001.
- [12] M. Kutter, "Watermarking Resisting to Translation, Rotation, and Scaling," *Proceedings of SPIE Vol. 3528*, pp. 423-431, 1998.
- [13] J.P. Parker, *Algorithms for Image Processing and Computer Vision*. John Wiley & Sons, Hoboken, NJ, 1996.



천 인 국

1983년 서울대학교 전자공학과 (공학사)

1985년 KAIST 전기및 전자공학과(공학석사)

1985년 3월 ~ 1988년 2월 삼성전자 종합연구소 연구원

1993년 KAIST 전기및 전자공학과(공학박사)

1993년 9월 ~ 현재 순천향대학교 정보기술공학부 부교수
관심분야: 영상처리, 컴퓨터비전, 인공지능, 디지털 콘텐츠