

차이값 히스토그램 쉬프팅과 오류 예측 보정을 이용한 가역 영상 워터마킹

(Reversible Image Watermarking with Differential Histogram Shifting and Error Prediction Compensation)

여 동 규[†] 이 해 연^{**} 김 병 만^{***} 김 경 수^{****}
 (Dong-Gyu Yeo) (Hae-Yeoun Lee) (Byeong Man Kim) (Kyung-Su Kim)

요약 가역 워터마킹 기법은 디지털 콘텐츠에 지각적 투명성을 유지하며 워터마크를 삽입한 후에, 이를 아무런 손상 없이 원본 상태로 복원할 수 있는 메시지 은닉 수단이다. 영상의 품질이 매우 중요한 의학, 군사, 원격측량, 예술작품 분야에서 워터마킹 응용은 영상처리와 분석과정에서 손상 없는 원본이 필요하기 때문에, 메시지를 검출하고 원본으로 복원하는 과정에서 어떠한 손상이라도 허용될 수 없는 완전한 가역성이 보장되어야 한다. 본 논문에서는 영상의 인접한 픽셀들 간의 차이값 히스토그램을 수정하여 메시지를 은닉하는 가역 영상 워터마킹 알고리즘을 제안한다. 제안한 방법에서는 높은 삽입용량과 지각적 투명성을 만족하기 위해 영상의 인접 픽셀들 간의 높은 유사성 특징을 이용하였다. 또한 오류 예측 보정 기법을 통하여 워터마크 삽입과정에서 발생할 수 있는 오버/언더플로우 문제와 salt-and-pepper 잡음 현상을 방지하였다. 제안한 가역 워터마킹 알고리즘의 성능을 검증하기 위하여 다양한 영상들에 대하여 기존 알고리즘들과 비교 분석하였다. 실험 결과에 따르면 제안한 알고리즘은 완전한 가역성과 함께 낮은 왜곡을 유지하면서도 높은 삽입용량을 얻을 수 있었다.

키워드 : 가역 워터마킹, 차이값 히스토그램, 히스토그램 쉬프팅, 오류 예측 보정

Abstract Reversible watermarking inserts watermark into digital media in such a way that visual transparency is preserved and then enables to restore the original media from the marked one, without any loss of media quality. This watermarking can be applied to quality-sensitive imaging such as medical imaging, military imaging, remote-sensing imaging, and precious artwork, where the original media should be preserved during image processing and analysis. In this paper, a reversible image watermarking technique that embeds message bits by modifying the differential histogram of adjacent pixels is presented. In order to satisfy both high embedding capacity and visual quality, the proposed technique exploits the fact that adjacent pixels in the image have highly spatial correlation. Also, we prevent overflow/underflow problem and salt-and-pepper artifacts by employing a predicted error compensation scheme. Through experiments using various test images, we prove that the presented technique provides perfect reversibility and high embedding capacity, while maintaining the induced-distortion low.

Key words : Reversible watermarking, Differential histogram, Histogram shifting, Predicted error compensation

· 본 연구는 문화체육관광부 및 한국문화콘텐츠진흥원의 2009년도 문화콘텐츠산업 기술지원사업의 연구결과로 수행되었음

논문접수 : 2009년 11월 18일
 심사완료 : 2010년 4월 8일

† 학생회원 : 국립금오공과대학교 컴퓨터공학부
 sylot@kumoh.ac.kr

** 정 회 원 : 국립금오공과대학교 컴퓨터공학부 교수
 haeyeoun.lee@kumoh.ac.kr
 (Corresponding author임)

*** 종신회원 : 국립금오공과대학교 컴퓨터공학부 교수
 bmkim@kumoh.ac.kr

**** 정 회 원 : KT 네트워크연구소
 kyungsu.kim@kt.com

Copyright©2010 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 소프트웨어 및 응용 제37권 제6호(2010.6)

1. 서론

데이터 은닉 기술은 음악, 영상, 동영상, 전자문서, 교육자료, 애니메이션과 같은 디지털 콘텐츠에 기밀 정보를 비가시적으로 삽입하는 기술로서, 소유권 증명, 저작권 보호, 방송 모니터링, 콘텐츠 인증 등의 다양한 목적으로 활용되고 있다. 암호화 기술도 디지털 콘텐츠 보호를 위한 방법이지만 그것은 콘텐츠 배포과정에서의 보호만 보장할 뿐이며, 한 번 복호화된 콘텐츠는 더 이상 보호될 수 없기 때문에 콘텐츠의 무결성을 입증하기 위한 충분한 수단을 제공하기에는 부족하다. 이에 반하여 데이터 은닉 기술은 응용에 따라 다양한 삽입용량과 시각적 투명성, 강인성, 기밀성, 계산 복잡도 등의 요구조건을 만족시킬 수 있다.

대표적인 데이터 은닉 기술인 디지털 워터마킹은 삽입될 메시지와 원본 콘텐츠가 밀접한 연관성을 가지고 있는데, 원본 콘텐츠에 대한 메타데이터 혹은 무결성 검증을 위한 인증코드, 저작권 정보 등이 워터마크의 내용이 될 수 있다. 워터마킹은 응용 관점에 따라 강인성(Robust) 워터마킹과 연성(Fragile) 워터마킹으로 분류된다 [1]. 강인성 워터마킹은 콘텐츠의 시각적 품질을 유지하면서 모든 가능한 왜곡 시도로부터 워터마크의 내용이 보호될 수 있도록 설계된다. 반면 연성 워터마킹은 아주 작은 변형만으로도 쉽게 워터마크가 손상되기 때문에 콘텐츠의 위조 및 변조에 대한 무결성 입증이나 인증에 유용하게 적용될 수 있다.

콘텐츠에 데이터를 은닉하려면 필연적으로 원본 콘텐츠의 수정이 불가피한데, 의료 영상이나 군사적 영상, 법률적 증거, 원격 측정값, 예술작품 등의 응용분야에서는 어떠한 손상도 없는 원본 영상이 필요하다. 변경의 정도가 극히 미미하고 인간의 지각능력으로는 전혀 알아볼 수 없을지라도 올바른 결정에 영향을 미칠 수 있으며 법률적 문제가 될 수 있기 때문이다. 연성 워터마킹의 한 종류인 가역(Reversible) 워터마킹은 워터마킹된 콘텐츠에서 메시지를 제거한 후 원본 콘텐츠로 완전한 복원이 가능하기 때문에 콘텐츠의 무결성 인증이나 위변조 조작에 대한 증명, 저작권 보호를 위한 용도로 이용되어질 수 있다[2].

본 논문에서는 영상 콘텐츠의 차이값 히스토그램을 수정하는 가역 영상 워터마킹 알고리즘을 제안한다. 높은 삽입용량과 시각적 투명성을 얻기 위하여 인접한 픽셀간의 공간적 유사도 특징을 이용하고, 오류 예측 보정 기법을 통하여 오버/언더플로우를 방지한다. 또한, 여러 영상 데이터들을 이용하여 실험하고 기존 연구들과 성능을 비교함으로써 제안한 알고리즘의 효율성을 입증한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서 가역

영상 워터마킹 분야의 기존 주요 연구들을 통하여 워터마킹의 알고리즘적인 차이에 대하여 논한다. 3장에서는 제안한 가역 워터마킹 알고리즘의 삽입 및 검출/복원 방법과 오류 예측 보정을 통한 오버/언더플로우 방지 기법에 대하여 설명한다. 실험 및 성능 분석은 4장에서 제시하며, 5장에서 결론을 맺는다.

2. 관련 연구

가역 워터마킹 알고리즘은 메시지를 삽입하여 수정된 영상이 원본 영상에 비해 큰 차이가 없도록 시각적으로 투명해야 하면서도 응용분야에서의 요구에 맞게 충분한 삽입용량을 제공해야 한다. 또한 상대적으로 자원 및 계산능력이 부족한 소형 장치에서도 응용이 가능하도록 계산 복잡도가 높지 않아야 하며, 메시지를 추출한 후 완전한 원본으로의 복원이 가능하도록 가역성을 보장해야 한다. 이러한 요구조건을 만족하기 위하여 여러 방법들이 연구되어졌는데, 이들은 각기 다른 영상의 특징을 이용하여 메시지를 삽입한다.

2.1 가역 워터마킹 알고리즘의 분류

메시지를 삽입하는 방법에 따라 기존 알고리즘들을 네 가지로 분류할 수 있다. 첫 번째 분류의 알고리즘들은 메시지를 삽입할 수 있는 공간을 확보하기 위해 영상의 특징을 이용한 압축 방법을 사용한다. Fridrich et al.[3]은 JBIG 무손실 압축 기법을 이용하여 비트 평면을 압축하고, 빈 공간에 메시지를 삽입했다. 그러나 하위 비트 평면은 노이즈 성분이 많아서 압축효율이 떨어지기 때문에 상대적으로 노이즈가 적은 상위 비트 평면에 메시지를 삽입하게 되어 시각적인 손상이 발생한다. Celik et al.[4]은 CALIC 무손실 압축 알고리즘을 적용하여 압축한 후 Generalized Least Significant Bit 삽입 기법으로 삽입용량을 높였다. 이들 방법의 성능은 영상의 구조적 특성과 사용된 압축 알고리즘의 성능에 따라 다르다.

두 번째 분류의 알고리즘들은 변환 영역에서 이루어지는 형태로서, DCT(Discrete Cosine Transform)나 DWT(Discrete Wavelet Transform) 등의 변환 계수에 메시지를 삽입한다. Yang et al.[5]은 블록 단위의 정수형 DCT 계수에 기반하여 여러 AC 계수를 선택하여 메시지를 삽입하였다. Xuan et al.[6]은 영상을 DWT로 변환 후 고주파 서브밴드에서 정수 웨이블릿 계수의 가운데 비트평면에 메시지 비트를 삽입했다. Lee et al.[7]은 영상의 블록에 정수대 정수 웨이블릿 변환을 적용하여 각 블록의 고주파 웨이블릿 계수에 메시지 비트를 삽입했다. 이들 방법은 변환 영역에서 이루어지기 때문에 계산 복잡도가 높다.

세 번째 분류의 알고리즘들은 차이값 확장 방법으로

서 원본 영상의 특성을 포함할 수 있는 작은 값을 생성하고 그 값을 확장함으로써 확장된 공간에 정보를 삽입하는 방법이다. Tian[8]은 정수형 웨이블릿 변환을 이용하여 영상의 차이값과 평균값을 이용한 특성 값을 계산하고 이 특성 값을 확장하여 메시지를 삽입하였다. 이 방법은 확장 가능한 모든 차이값들의 위치정보 맵을 워터마크와 함께 삽입해야하는 부담이 있다. Alattar[9]는 임의의 정수 변환에 적용 가능한 정형화된 차이값 확장 방법을 제안하였다. Kamstra와 Heijmans[10]는 어느 위치가 확장 가능한지 예측하는 기법을 적용하여 성능을 향상시켰다. Thodi와 Rodriguez[11]는 위치정보 맵을 삽입하기 위해 히스토그램을 쉬프팅하는 방법과, 이웃하는 픽셀들의 공간적 유사도를 활용한 예측 오류 확장 방법을 제안하였다.

네 번째 분류인 히스토그램 수정 방법은 영상의 히스토그램을 이용한다. 대부분의 히스토그램 기반 방법들은 모든 처리과정이 공간 영역에서 수행되어지기 때문에 변환 영역에서의 방법에 비해 계산 복잡도가 훨씬 낮다. 이 방법은 알고리즘에 따라서 원 영상의 픽셀값 히스토그램이나 차분 영상의 히스토그램을 이용한다. Ni et al.[12]과 Versaki et al.[13]의 방법은 히스토그램의 최소점과 최대점을 이용하는데 삽입용량은 최대점(Peak)에 속한 픽셀의 빈도수에 의해 정해진다. Hwang et al.[14]과 Kuo et al.[15]은 Ni의 방법을 확장하여 최소점과 최대점에 대한 정보를 저장하는 위치정보 맵을 이용하였다. Lin et al.[16]은 영상을 서로 겹치지 않는 블록으로 나누고 각 블록에서의 차분 영상을 생성한 후, 각각의 차분 영상에 대한 히스토그램을 수정하여 메시지를 삽입하였다. Tsai et al.[17]은 각 블록에서 기준 픽셀과 나머지 픽셀과의 차이값으로 구성된 차분 영상을 이용하였다. 비록 Lin의 방법과 Tsai의 방법이 기존 히스토그램 기반의 방법보다 높은 삽입용량을 보이지만, 모든 블록들의 Peak에 대한 정보도 함께 전송해야만 했다. 따라서 삽입용량을 증가시킬수록 오버헤드 정보, 즉 Peak 정보의 양도 증가하여 실제 유효 삽입용량이 그만큼 줄어들게 된다.

Lee et al.[18]은 기존 알고리즘들과 유사하게 영상의 홀수 라인과 짝수 라인과의 차분 영상을 이용하여 히스토그램 계산하였으며, 다만, 오버 및 언더플로우 방지를 위하여 0과 255 밝기값을 1과 254로 대체한 후 위치정보 맵을 이용하여 표시하였다. 이 방법은 낮은 삽입용량에서는 픽셀들의 지역적 유사성이 유지되어 높은 품질을 보이지만, 삽입용량을 높이기 위하여 반복적인 삽입을 수행할수록, 차이값 히스토그램 계산 방법의 문제로 픽셀들의 지역적 유사성이 심각히 왜곡되어 삽입용량의 변화가 적고, 반복횟수만 증가하여, 영상의 품질이 급격

하게 저하되는 단점을 갖는다.

최근에 Kim et al.[19]은 영상의 공간적 유사성을 더욱 활용하기 위하여 영상을 일정한 간격으로 샘플링한 서브영상으로 나누고, 기준이 되는 서브영상과 나머지 서브영상들과의 차이값 히스토그램을 생성한 후, 각 히스토그램을 쉬프팅하여 메시지를 삽입하였다. 삽입용량을 증가시키기 위해서는 가능한 높은 최대점을 확보해야 하는데, 인접한 픽셀들간의 차이가 크지 않다는 영상의 지역성을 활용하여 좋은 성능을 보여주었다. 특히 지역성이 높아서 고주파 성분보다는 저주파 성분이 더 많은 영상에 대해 품질 대비 삽입용량 척도에서 매우 효율적이다. 일반적인 영상들이 저주파 성분이 더 높다는 특성을 감안하면 다양한 용도로 활용되기에 적합한 방법이다. 또한 이 방법은 차이값 히스토그램의 특성상 최대점이 0인 경우가 대부분이라는 가정하에 최대점의 위치를 0으로 고정시켰기 때문에 최대점과 최소점의 위치에 대한 오버헤드 정보가 필요하지 않게 되었으며, 다만 서브샘플링 간격과 삽입레벨 정보를 위한 12 비트만의 오버헤드가 발생했다. 삽입용량은 히스토그램에서 쉬프팅할 영역의 크기를 지정하는 삽입레벨 변수를 통해 조절할 수 있기 때문에, 응용의 요구에 따라 유연하게 적용될 수 있다. 그러나 서브영상으로 나누어지지 않는 나머지 영역과 더불어 한 개의 기준 서브영상은 수정하기 않기 때문에 영상의 일부 공간은 메시지 삽입에 이용할 수 없으며, 메시지가 삽입된 영상에 재삽입(Multi-round)을 하는 경우 수정되지 않은 영역 때문에 지역성이 낮아진다는 단점이 있다.

2.2 오버/언더플로우 방지

히스토그램 수정 방법을 통한 가역 워터마킹에서는 오버/언더플로우 문제를 고려해야 한다. 8 비트 회색조 영상의 경우, 히스토그램이 쉬프팅될 때 0과 255에 가까운 픽셀들에서 오버/언더플로우가 발생할 수 있다. Honsinger et al.[20]은 modulo-256 방법을 통해 255를 초과하는 픽셀은 0으로 돌아가서 증가하고, 0 미만으로 감소하는 픽셀은 255로 돌아가서 감소하도록 하였다. modulo-256 방법이 오버/언더플로우를 해결하긴 하지만, 변경된 픽셀값과 주변 픽셀값들과의 차이가 크게 되는 salt-and-pepper 잡음 현상이 발생하게 된다.

Fridrich et al.[3]은 이 잡음 문제를 줄이기 위하여 cyclic-modulo 방법을 제안하였다. modulo 연산의 간격을 하나의 구간이 아니라, 여러 개의 구간으로 나누는 방법이다. 예를 들어 64 cyclic-modulo는 64개씩 4개의 구간으로 나누어서 픽셀값들을 증감하는 것이다. cyclic-modulo 방법이 salt-and-pepper 잡음을 줄여주지만 그 변화의 폭을 cycle 크기만큼 줄였을 뿐이며 완전한 해결책은 아니다. 또한 modulo 방법이 많은 영상에 대해

여 좋은 성능을 보이기는 하지만 삽입된 메시지를 추출한 후 원 영상으로 복원할 때 modulo 연산이 적용된 픽셀인지 아닌지를 구분해야만 한다. 기존의 연구들에서는 일반적인 영상의 경우 인접한 픽셀간의 차이가 크게 나지 않는다는 특성을 이용하여 임계값을 설정하고, 그 값 이상으로 차이나는 픽셀에 대하여 modulo 연산이 적용되었다고 판단한다. 따라서 픽셀 밝기 변화가 매우 심한 영상에 대해서는 적용하기 어렵다.

오버/언더플로우 문제를 해결하기 위한 다른 방법은 위치정보 맵을 이용하는 방법이다[8]. 오버/언더플로우가 발생한 픽셀의 위치를 표시한 맵을 사용하여 삽입 과정에서 제외시키거나, 또는 복원할 때 역으로 modulo 연산을 하여 보정하는 방법이다. 맵을 이용하는 방법이 비록 완전한 가역성을 보장하기는 하지만, 여전히 맵 정보를 전달해야 하는 오버헤드를 안고 있기 때문에 실제 유효 삽입용량이 줄어든다. 본 논문에서는 오버/언더플로우를 해결하기 위하여 오버/언더플로우가 발생가능한 픽셀을 예측하여 미리 백쉬프팅시킨 후에 메시지를 삽입하도록 하는 오류 예측 보정 방법을 활용하였다.

3. 제안하는 가역 워터마킹 알고리즘

본 논문에서는 원본 영상을 완전하게 복원할 수 있도록 메시지를 삽입하기 위해서 인접 픽셀간의 차이값을 이용한 히스토그램 쉬프팅 기법을 이용한다. 또한 메시지를 삽입할 때 오버/언더플로우가 발생 가능한 픽셀들에 대해서는 오류 예측 보정을 통하여 오버/언더플로우가 발생되지 않도록 한다. 먼저 오류를 예측하여 원영상을 보정하고, 보정된 영상에 대해 차이값 히스토그램을 생성한 후, 히스토그램을 수정하여 메시지를 삽입한다. 메시지 검출 및 영상 복원과정에서는 차이값 히스토그램을 역으로 쉬프트시키는 방법을 이용하여 수행되며, 오류 예측 정보를 이용하여 최종 영상으로 복원시킨다. 오류 예측 정보는 압축된 맵의 형태로 메시지에 포함되어 삽입되어진다.

3.1 차이값 히스토그램

히스토그램 쉬프팅 기법을 이용하는 가역 워터마킹 알고리즘들은 데이터를 삽입하려는 최대점 주위의 밝기 값 히스토그램을 수정하여 빈 공간을 확보하고 확보된 공간에 최대점을 분산시키는 방법을 사용하여 데이터를 삽입한다. 따라서 높은 데이터 삽입용량을 얻기 위해서는 최대점들을 많이 확보해야 한다. 하지만, 많은 최대점들을 사용할수록 원본 영상의 왜곡이 심해지고 오버헤드 정보의 양과 알고리즘의 복잡도가 증가한다. 반면 차이값 히스토그램을 이용하면 하나의 최대점만 사용하더라도 높은 삽입용량을 얻을 수 있다.

차이값 히스토그램은 인접한 픽셀과의 밝기값 차이를 이용하여 계산한다. 그림 1과 같이 Lena 영상에서 일반 밝기값 히스토그램의 최대값은 2,751이나 차이값 히스토그램의 최대값은 22,252로 최대점의 높이 차이가 큰 것을 알 수 있다. 높은 최대점을 갖는 것은 적은 왜곡으로 높은 삽입용량을 얻을 수 있다는 것을 의미하므로, 차이값 히스토그램을 이용하는 것이 더 유리하다. 또한 일반 히스토그램을 이용한 방법은 데이터 삽입 후 최대점의 위치가 변하기 때문에 원래의 최대점의 위치를 기억하기 위한 추가적인 오버헤드가 발생한다. 하지만 차이값 히스토그램을 이용한 방법은 인접한 픽셀간의 변화가 작다는 영상 데이터의 지역성 특징으로 인하여 차이값들이 0 주변으로 몰려있기 때문에 최대점의 위치를 고정시켜서 처리할 수 있으므로 삽입위치 정보에 대한 오버헤드가 필요하지 않다는 장점을 가진다.

차이값 히스토그램에서 최대점의 높이는 차이값을 얻기 위한 방법에 따라 달라진다. 그림 2의 단순한 영상의 예를 통해 설명한다. 기존의 알고리즘들은 (1,2), (3,4), (5,6), (7,8)의 쌍에 대하여 차이값을 계산하고 메시지를 삽입한 후 다시 (2,3), (4,5), (6,7) 쌍에 대하여 계산하기 때문에 지역성이 저하되는 단점이 있다.

본 논문에서는 기존의 알고리즘과는 달리 (1,2), (2,3), (3,4), (4,5), (5,6)의 순서로 차이값을 계산하고, 메시지

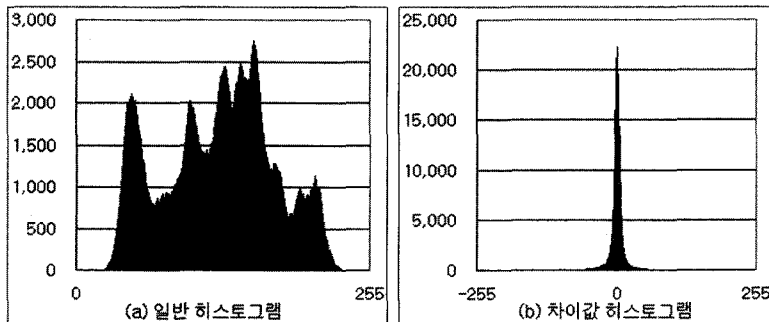


그림 1 Lena 영상의 히스토그램

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18

그림 2 차이값 히스토그램을 얻기 위한 영상(3×6)의 예가 삽입된 영상을 만들어 낼 때 원본영상을 이용하는 방법을 사용하여 차이값 수정분의 누적을 피하였다. 메시지를 추출하는 과정에서도 한 픽셀씩 점진적으로 복원하여 완전한 가역성을 얻어내었다.

3.2 오류 예측 보정

메시지를 삽입하기 위한 공간을 마련하기 위하여 히스토그램을 쉬프트하면 영상의 최소 밝기값과 최대 밝기값에 가까운 픽셀들에서 오버/언더플로우가 발생한다. 본 논문에서는 오버/언더플로우를 방지하기 위하여 최소값과 최대값에 가까운 픽셀들, 즉 바운드 영역에 포함되는 픽셀들을 예측하여 바운드에서 멀어지도록 백쉬프팅 시킴으로써 오버/언더플로우를 방지한다. 바운드 영역의 범위는 삽입레벨에 따라 달라진다.

백쉬프팅(Back Shifting)된 픽셀들은 맵을 구성하여 위치를 기억하며, 맵은 압축된 후 삽입할 워터마크에 포함되어 수신측에 전송된다. 메시지를 삽입할 때는 바운드 영역을 백쉬프팅시킨 후, 맵과는 무관하게 삽입과정이 수행되며, 또한 검출 및 복원과정에서도 모든 과정을 마친 후에 맵을 참조하여 픽셀값을 보정하므로, 삽입 및 검출과정 중에는 맵을 참조할 필요가 없다. 이를 통해 salt-and-pepper 잡음 현상도 완전하게 해결할 수 있다. 그림 3은 8-bit 영상에서 삽입레벨 1을 위해 바운드 영역을 백쉬프팅시키는 예를 나타낸다.

본 논문에서는 백쉬프팅된 맵 데이터의 용량을 줄이기 위하여 무손실 영상 압축에서 좋은 성능을 나타내는 JBIG1 압축 기법을 이용하였다. JBIG1 압축은 팩시밀리

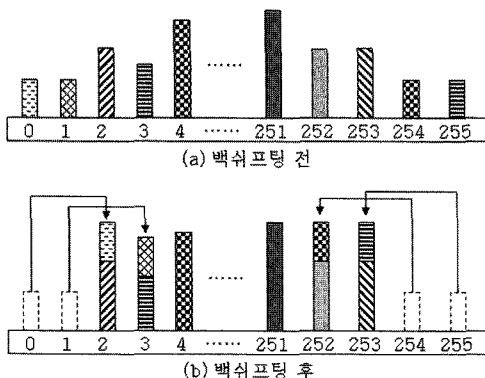


그림 3 8-bit 영상에서 바운드 백쉬프팅 예(삽입레벨 L=1의 경우)

응용과 같은 bi-level 영상에서 더 좋은 성능을 보인다.

원본 영상과 동일한 크기의 맵은 0과 1로만 구성되어 있다. 이 맵을 압축하기 위하여 바이트 단위로 구성된 맵을 비트단위의 맵으로 변환하고, 다시 JBIG1 압축과정을 거치게 된다. 예를 들면, 8-bit 512×512 영상의 크기는 262,144 바이트이고, 이 영상에 대한 비트단위의 맵은 32,768 Bytes 크기로 변환되며 다시 JBIG1 압축을 수행하면 삽입 레벨에 따라 달라지지만, 실험에서 사용한 영상들의 경우 31~380 바이트로 압축되어 평균 99.4%의 높은 압축효율을 보였다.

3.3 워터마크 삽입 알고리즘

메시지는 차이값 히스토그램을 수정하여 삽입하게 되며, 삽입되는 메시지는 균등 분포로 가정한다. 삽입되는 메시지의 용량 및 워터마크 영상의 품질은 응용분야의 요구에 따라 삽입레벨 L로 조절한다. 메시지 삽입은 다음 5 단계의 과정을 거치게 되며, 그림 4는 전체 삽입과정을 도시하였다.

단계 0 : 원본 영상을 I에 저장한다. 본 논문에서는 2 차원인 영상을 1 차원 배열에 저장하여 처리한다.

단계 1 : 오버/언더플로우가 발생할 수 있는 바운드 영역을 예측하여 백쉬프팅한다. 바운드 영역의 크기는 삽입레벨 L에 따라 달라지며, 픽셀의 최소 밝기값과 최대값에서 각각 L 크기만큼씩이 해당된다. 백쉬프팅된 픽셀들의 위치는 원본 영상과 동일한 크기의 맵(MAP)을 구성하여 저장하는데, 백쉬프팅되었으면 1, 아니면 0으로 표시하며 다음 식 (1)로 구해진다. M과 N은 각각 영상의 가로와 세로 크기이다.

$$MAP_{(i)} = \begin{cases} 1, & \text{if } 0 \leq I_{(i)} \leq L \\ 1, & \text{if } (UPPER-L) \leq I_{(i)} \leq UPPER \\ 0, & \text{otherwise} \end{cases}$$

for $0 \leq i \leq (M \times N - 1)$, $UPPER = 2^n - 1$, $n = \text{Bit Depth}$. (1)

원본 영상으로부터 바운드 영역이 백쉬프팅된 영상은 다음 식 (2)를 통하여 얻어진다.

$$Is_{(i)} = \begin{cases} I_{(i)} + (L+1), & \text{if } 0 \leq I_{(i)} \leq L \\ I_{(i)} - (L+1), & \text{if } (UPPER-L) \leq I_{(i)} \leq UPPER \\ I_{(i)}, & \text{otherwise} \end{cases}$$

for $0 \leq i \leq (M \times N - 1)$, $UPPER = 2^n - 1$, $n = \text{Bit Depth}$. (2)

단계 2 : 수정된 영상으로부터 차이값 벡터 D를 구한다. 벡터의 첫 값은 영상의 첫 픽셀값으로 설정되고, 이후는 바로 앞 픽셀과의 차이값으로 구성된다.

$$D_{(i)} = \begin{cases} Is_{(0)}, & \text{if } i = 0 \\ Is_{(i)} - Is_{(i-1)}, & \text{otherwise} \end{cases}$$

for $0 \leq i \leq (M \times N - 1)$. (3)

단계 3 : 차이값 벡터로부터 차이값 히스토그램 H를 구한다음, 메시지 삽입을 위한 공간을 마련하기 위해 삽

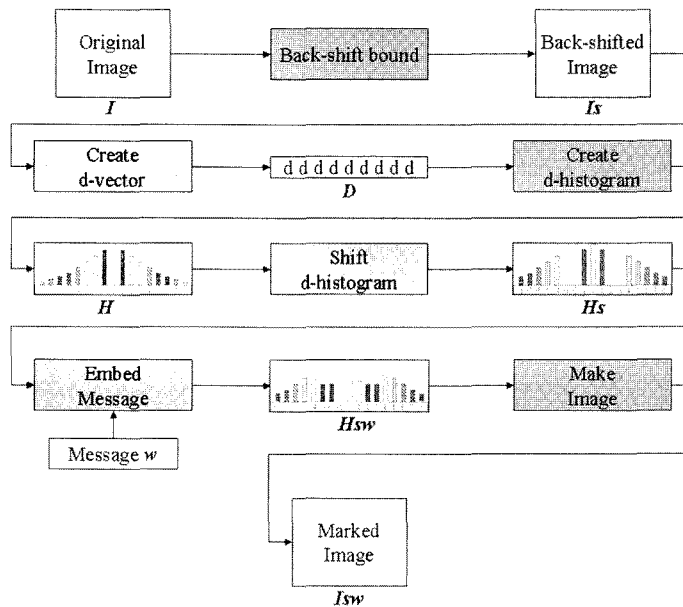


그림 4 워터마크 삽입 절차

입레벨에 따라 다음과 같이 히스토그램을 쉬프트시킨다.

$$H_s = \begin{cases} H + (L + 1), & \text{if } H \geq (L + 1) \\ H - (L + 1), & \text{if } H \leq -(L + 1) \end{cases} \quad (4)$$

히스토그램 쉬프팅은 다음 식 (5)와 같이 차이값 벡터를 수정하여 이루어진다.

$$D_{s(i)} = \begin{cases} D_{(0)}, & \text{if } i = 0 \\ D_{(i)} + (L + 1), & \text{if } D_{(i)} \geq (L + 1) \\ D_{(i)} - (L + 1), & \text{if } D_{(i)} \leq -(L + 1) \end{cases} \quad (5)$$

for $0 \leq i \leq (M \times N - 1)$.

단계 4 : 히스토그램을 수정하여 메시지 $w(n)$ 를 삽입하는 작업은 다음 식 (6)과 같이 차이값 벡터를 수정하여 이루어진다.

$$D_{sw(i)} = \begin{cases} D_{s(0)}, & \text{if } i = 0 \\ 2D_{s(i)}, & \text{if } -L \leq D_{s(i)} \leq -1 \text{ and } w(n) = 0 \\ 2D_{s(i)} - 1, & \text{if } -L \leq D_{s(i)} \leq -1 \text{ and } w(n) = 1 \\ 2D_{s(i)}, & \text{if } 0 \leq D_{s(i)} \leq L \text{ and } w(n) = 0 \\ 2D_{s(i)} + 1, & \text{if } 0 \leq D_{s(i)} \leq L \text{ and } w(n) = 1 \end{cases} \quad (6)$$

for $0 \leq i \leq (M \times N - 1)$, where $w(n) \in \{0, 1\}$.

단계 5 : 메시지가 삽입된 수정된 히스토그램으로부터 메시지가 삽입된 최종 영상을 구하는 절차는 다음 식을 통하여 이루어진다.

$$I_{sw(i)} = \begin{cases} D_{sw(0)}, & \text{if } i = 0 \\ I_{s(i-1)} + D_{sw(i)}, & \text{otherwise} \end{cases} \quad (7)$$

for $0 \leq i \leq (M \times N - 1)$.

차이값 벡터는 메시지가 삽입되어 쉬프트가 되어있는 상태이기 때문에 차이값 벡터만을 이용하여 영상을 만들게 되면 앞 픽셀들의 쉬프트된 누적분의 영향으로 울

바른 영상을 얻을 수 없음을 주의해야 한다.

삽입레벨 $L=1$ 에서 히스토그램을 수정하여 메시지를 삽입하는 과정의 예를 그림 5에 나타내었다. 설명을 위해 그림 5(c)와 그림 5(d)는 각각 별도로 나타내었지만, 한 번의 차이값 벡터 스캔 과정을 거치며 수행된다. 삽입과정이 완료된 히스토그램에는 -1 공간이 비워져 있는데, 이는 영상을 복원하는 측에서 공격으로 인한 영상의 훼손 여부를 쉽게 확인하기 위함이다. 만일 히스토그램에서 -1에 해당하는 빈도수가 하나라도 나타나게 되면 나머지의 복원과정을 거치지 않고도 영상이 훼손되었음을 쉽게 확인할 수 있게 된다.

본 논문에서 제안한 방법은 메시지가 은닉된 영상과 함께 몇 가지의 오버헤드 정보를 같이 수신측에 전달해야 한다. 수신측에서 복원을 위해 필요한 추가적인 정보는 삽입레벨 L 과, 맵의 크기, 맵 데이터이다. 삽입레벨 L 은 수신측과 공유된 비밀키를 이용하여 선택된 임의의 픽셀의 LSB로 숨겨진다. 맵 데이터의 크기를 줄이기 위하여 3.2절에서 설명한 JBIG1 방법을 사용하여 압축하였으며, 삽입레벨을 제외한 오버헤드는 메시지의 앞부분에 위치하여 삽입된다.

3.4 워터마크 검출 및 복원 알고리즘

삽입된 메시지를 검출하기 위하여 수신측에서는 다음 6 단계의 과정을 거쳐 영상을 복원하게 된다. 차이값 히스토그램의 특성상 픽셀을 복원하기 위해서는 앞 픽셀뿐만 아니라 앞 픽셀과의 차이값 또한 같이 복원해야 하므로, 복원 과정은 하나의 단계씩 완료되는 것이 아니라

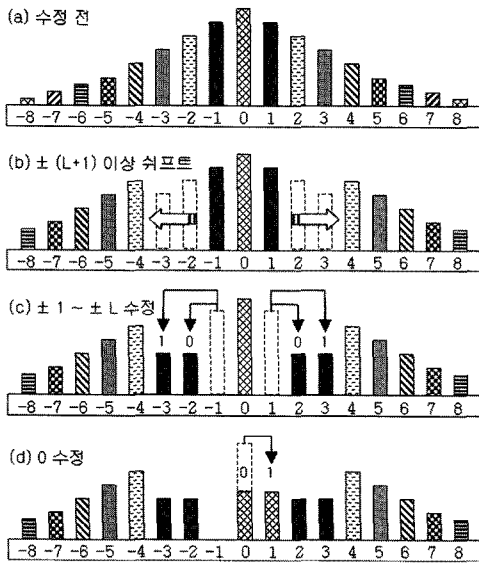


그림 5 메시지 삽입과정에서의 히스토그램 수정 절차 (삽입레벨 $L=1$ 의 경우)

1~5 단계의 과정을 한 픽셀씩 반복하며 수행해야 한다. 히스토그램의 -1 공간은 영상의 훼손여부를 인증하기 위한 방법이므로, 복원 과정중 -1의 차이값이 나타나면 복원을 중단한다. 수신측은 미리 공유된 비밀키를 이용하여 LSB로부터 삽입레벨 L 을 먼저 알아내고 나머지 과정을 진행한다. 전체 검출 및 복원 과정을 그림 6에 도시하였다.

단계 0 : 메시지가 삽입된 영상을 I_{sw} 에 저장하며, 삽입과정에서와 마찬가지로 1 차원 배열에 저장하여 관리한다. 비밀키를 이용하여 LSB에 숨겨진 삽입레벨 L 값을 알아낸다.

단계 1 : 차이값 벡터를 다음 수식을 통하여 구한다. 복원된 영상인 I_s 의 첫 번째 값을 의미하는 $I_{s(0)}$ 는 삽입과정에서 수정되지 않은 픽셀값인 $I_{sw(0)}$ 의 값으로 결정된다.

$$Dsw_{(i)} = \begin{cases} I_{sw(0)}, & \text{if } i=0 \\ I_{sw(i)} - I_{s(i-1)}, & \text{otherwise} \end{cases} \quad (8)$$

for $0 \leq i \leq (M \times N - 1)$.

단계 2 : 계산된 차이값을 이용하여 삽입된 메시지를 검출한다. 메시지가 삽입된 공간이 아니면 다음 단계로 넘어간다.

$$w(n) = \begin{cases} 0, & \text{if } -(2L+1) \leq Dsw_{(i)} \leq -2 \text{ and } \text{mod}(Dsw_{(i)}, 2) = 0 \\ 1, & \text{if } -(2L+1) \leq Dsw_{(i)} \leq -2 \text{ and } \text{mod}(Dsw_{(i)}, 2) = -1 \\ 0, & \text{if } 0 \leq Dsw_{(i)} \leq (2L+1) \text{ and } \text{mod}(Dsw_{(i)}, 2) = 0 \\ 1, & \text{if } 0 \leq Dsw_{(i)} \leq (2L+1) \text{ and } \text{mod}(Dsw_{(i)}, 2) = 1 \end{cases} \quad (9)$$

for $1 \leq i \leq (M \times N - 1)$.

단계 3 : 삽입된 메시지를 검출하였으면 다음 식 (10)을 이용하여 메시지를 제거한다. 메시지가 삽입된 공간이 아니면 다음 단계로 넘어간다.

$$D_{s(i)} = \begin{cases} Dsw_{(0)}, & \text{if } i=0 \\ \left\lfloor \frac{Dsw_{(i)}}{2} \right\rfloor, & \text{if } -(2L+1) \leq Dsw_{(i)} \leq (2L+1) \\ Dsw_{(i)}, & \text{otherwise} \end{cases}$$

for $0 \leq i \leq (M \times N - 1)$.

(10)

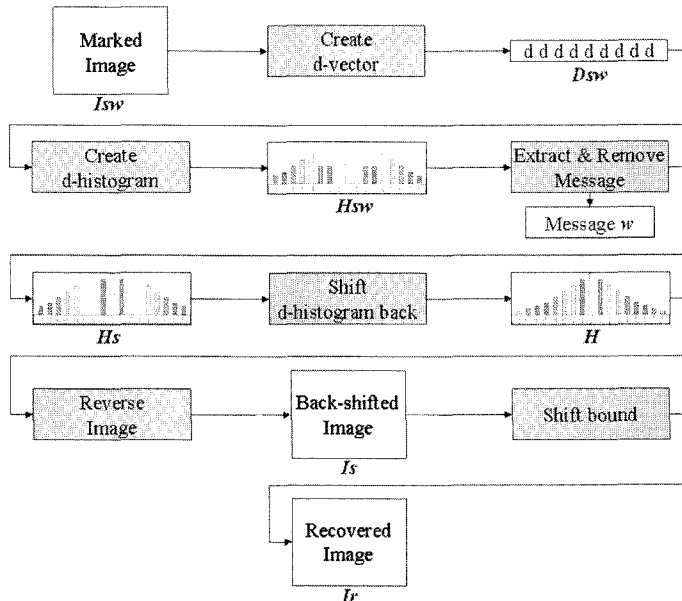


그림 6 워터마크 검출 및 복원 절차

단계 4 : 메시지를 삽입하기 위한 공간을 마련하기 위하여 쉬프트하였던 차이값 히스토그램을 원위치로 복구시킨다.

$$H = \begin{cases} H_s - (L+1), & \text{if } H_s \geq (2L+2) \\ H_s + (L+1), & \text{if } H_s \leq -(2L+2) \end{cases} \quad (11)$$

이 과정은 다음 식 (12)와 같이 차이값 벡터를 수정함으로써 수행된다.

$$D_{(i)} = \begin{cases} D_{s(0)}, & \text{if } i = 0 \\ D_{s(i)} - (L+1), & \text{if } D_{s(i)} \geq (2L+2) \\ D_{s(i)} + (L+1), & \text{if } D_{s(i)} \leq -(2L+2) \\ D_{s(i)}, & \text{otherwise} \end{cases} \quad (12)$$

for $0 \leq i \leq (M \times N - 1)$.

단계 5 : 복원된 앞 픽셀과 복원된 차이값을 이용하여 원 영상을 복원한다.

$$I_{s(i)} = \begin{cases} I_{sw(0)}, & \text{if } i = 0 \\ I_{s(i-1)} + D_{(i)}, & \text{otherwise} \end{cases} \quad (13)$$

for $0 \leq i \leq (M \times N - 1)$.

단계 6 : 복원된 영상을 검출된 메시지로부터 분리해 낸 백쉬프팅 맵 MAP을 참조하여 바운드 영역을 원래의 위치로 쉬프트시켜서 최종 복원 영상을 만들어낸다.

$$I_{r(i)} = \begin{cases} I_{s(i)}, & \text{if } MAP_{(i)} = 0 \\ I_{s(i)} - (L+1), & \text{if } MAP_{(i)} = 1 \text{ and } I_{s(i)} < \frac{UPPER}{2} \\ I_{s(i)} + (L+1), & \text{if } MAP_{(i)} = 1 \text{ and } I_{s(i)} > \frac{UPPER}{2} \end{cases}$$

for $0 \leq i \leq (M \times N - 1)$, $UPPER = 2^n - 1$, $n = \text{BitDepth}$. (14)

3.1절의 그림 2를 통해 설명한 바와 같이, 기존의 알고리즘들은 모든 픽셀에 대하여 한 단계씩 완료하며 병렬적으로 복원할 수 있지만, 본 논문에서 사용한 차이값 계산 방법은 (1,2), (2,3), (3,4), (4,5)의 순서이기 때문에 영상을 복원하려면 현재의 픽셀이 완전하게 복원되어야 다음 픽셀을 복원할 수 있다. 따라서 한 픽셀씩 차이값을 계산하면서 점진적으로 복원하여야하므로 단계 1~5를 한 픽셀마다 반복 수행해야 한다. 메시지를 복원하는 과정에서 히스토그램이 수정되는 과정을 그림 7에 나타내었다. 삽입할 때와 마찬가지로 그림 7(b)와 그림 7(c)는 이해를 돕기 위하여 각각 별도로 나타내었지만, 한번의 차이값 벡터 스캔 과정을 거치며 수행된다.

4. 실험 및 성능 평가

본 논문에서 실험에 사용된 영상은 8-bit 회색조 512x512 영상 7개이며 그림 8에 나타내었다. 삽입레벨 L 은 0~9까지에 대하여 실험하였으며 삽입되는 메시지의 내용은 0과 1로 구성되며 균등 분포를 따른다. 워터마크 메시지에 포함되는 오버헤드 정보는 맵의 크기정보가 32 비트(4 바이트)이며, 압축된 맵 데이터는 삽입레벨에 따라 달라진다. 삽입레벨 L 은 공유된 비밀키에 의

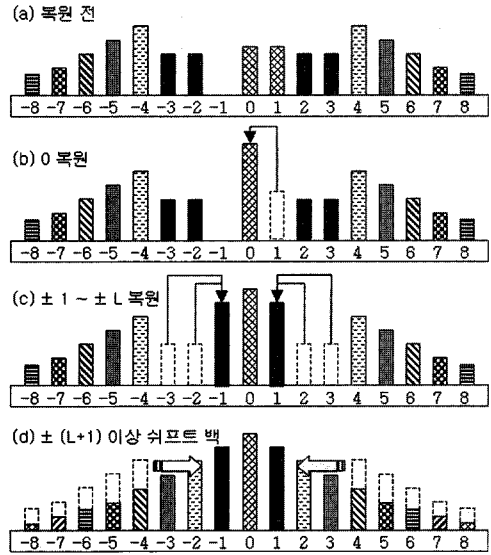


그림 7 메시지 검출과정에서의 히스토그램 수정 절차 (삽입레벨 $L = 1$ 의 경우)

해 LSB를 선택하여 삽입되는 방식이므로 영상에 수정을 가하지 않기 때문에 오버헤드 정보에 포함되지 않는다.

표 1에 실험을 통하여 확인한 압축된 맵의 크기를 나타내었다. O 혹은 U 값은 바운드 백쉬프팅 기법을 적용하지 않았을 때 실제로 발생한 오버/언더플로우 개수이며, B는 삽입레벨에 따라 오류를 예측하여 백쉬프팅한 바운드 영역의 픽셀 개수이다. 32,768 바이트 크기의 맵을 JBIG1 기법으로 압축한 크기를 바이트 단위로 S에 나타내었다. 표 1에 언급하지 않은 Lena, Airplane, Tank 영상은 오버/언더플로우가 발생하지 않았기 때문에 맵 데이터가 메시지에 삽입되지 않는다. 성능평가를 위해 다음에서의 모든 실험결과들은 삽입가능용량에서 오버헤드의 용량을 차감한 유효 삽입용량으로 계산되어졌다.

4.1 실험 결과

본 논문에서 실험한 7개의 영상에 대하여 0~9의 삽입레벨로 메시지를 삽입한 결과를 표 2와 표 3에 나타내었다. 유효삽입용량은 오버헤드 정보를 차감한 용량이며, 영상의 품질은 PSNR(dB)로 측정하였다.

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (p_{(i,j)} - p'_{(i,j)})^2 \quad (15)$$

$$PSNR_{dB} = 10 \log_{10} \frac{(2^n - 1)^2}{MSE} \quad (16)$$

M 과 N 은 각각 영상의 가로 및 세로 크기이며, $p_{(i,j)}$ 는 원본 영상의 픽셀값, $p'_{(i,j)}$ 는 마크가 삽입된 영상의 픽셀값이고, n 은 한 픽셀을 표현할 때 필요한 비트의 수이다. 그림 9는 실험결과를 삽입레벨에 대하여 삽입용량을

표 1 압축된 맵의 크기

삽입레벨	Baboon			Boat			Aerial			Trucks		
	U	B	S	U	B	S	O	B	S	O	B	S
0	5	39	38	1	1	31	247	512	32	-	-	-
1	13	58	44	1	1	31	259	512	32	-	-	-
2	14	86	51	1	1	31	252	512	32	-	-	-
3	13	130	55	1	1	31	266	519	55	-	-	-
4	17	173	58	1	1	31	296	555	119	-	-	-
5	18	206	56	1	1	31	321	589	157	-	-	-
6	22	226	57	1	1	31	336	624	201	1	1	34
7	24	235	62	2	2	37	380	689	274	1	1	34
8	30	246	72	2	2	37	401	752	329	1	1	34
9	30	254	84	2	2	37	430	810	380	1	1	34

O/U : 오버/언더플로우 발생픽셀수, B : 오류예측보정 픽셀수, S : 압축된 맵의 크기(Bytes)

표 2 삽입레벨에 따른 품질(PSNR: dB)

삽입레벨	Lena	Baboon	Boat	Airplane	Aerial	Tank	Trucks
0	48.32	48.20	48.33	48.48	48.33	48.42	48.30
1	42.73	42.34	42.78	43.21	42.74	42.69	42.52
2	39.73	39.01	39.83	40.54	39.70	39.52	39.27
3	37.77	36.71	37.90	38.82	37.65	37.46	37.08
4	36.36	34.97	36.50	37.60	36.12	35.95	35.45
5	35.30	33.59	35.42	36.65	34.90	34.81	34.17
6	34.46	32.46	34.54	35.88	33.90	33.92	33.14
7	33.77	31.50	33.80	35.23	33.04	33.20	32.29
8	33.20	30.68	33.17	34.66	32.31	32.62	31.57
9	32.71	29.96	32.61	34.17	31.66	32.14	30.96

표 3 삽입레벨에 따른 삽입용량(Payload : bits)

삽입레벨	Lena	Baboon	Boat	Airplane	Aerial	Tank	Trucks
0	22,220	8,108	23,490	40,025	23,454	33,752	19,858
1	62,978	24,507	68,022	102,847	63,627	42,123	35,862
2	98,905	40,559	105,647	144,202	93,626	87,570	63,023
3	129,195	56,114	134,822	172,041	115,807	106,931	83,520
4	153,288	70,706	156,077	190,392	132,448	133,620	102,151
5	172,171	84,237	171,785	202,626	145,360	155,470	120,612
6	186,759	96,943	183,403	211,155	156,054	171,034	134,714
7	198,262	108,733	192,131	217,878	164,597	188,005	149,237
8	207,039	119,291	199,459	222,977	171,867	198,057	161,731
9	213,989	128,970	205,402	226,979	178,388	210,792	173,430

그래프로 나타낸 것이고, 그림 10은 삽입레벨에 따른 품질을 나타낸 것이다. 그림 11에는 실험결과를 바탕으로 다양한 삽입용량에 따른 품질을 나타내었다. 이 결과를 통해 알 수 있듯이, 제한한 워터마킹 알고리즘의 성능은 영상 특성에 따라 달라진다. 예를 들면, Airplane이나 Boat처럼 저주파 성분을 더 많이 포함하고 있는 영상이 품질을 좋게 유지하면서도 높은 삽입용량을 얻는다는 것을 알 수 있다. 그림 8은 원본 영상과 함께 여러 삽입레벨로 메시지가 삽입된 영상을 보여준다. 그림에서 알 수 있듯이 적절한 삽입용량을 보장하면서도 메시지가 삽입된 영상의 시각적 품질이 만족할만한 수준의 결과

를 얻을 수 있다.

4.2 성능 비교

표 4에 Lena, Baboon, Boat, Airplane 영상에 대하여 기존의 히스토그램 기반 알고리즘들과의[12-15,17,19] 성능 비교를 요약하였다. Tsai et al.[17]의 실험에서는 블록의 크기를 4×4로 설정하였으며, Kim et al.[19]의 실험에서는 Δu , Δv , L 변수를 각각 3, 3, 0으로 설정하였다. Tsai et al.의 결과에서 “N/A(Not Applicable)”은 Baboon 영상이 고주파 성분이 많이 포함되어 있는 특성으로 인해 메시지를 삽입할 수 있는 충분한 Peak가 없어서 유효삽입용량이 (-) 결과를 보이기 때문에 적용

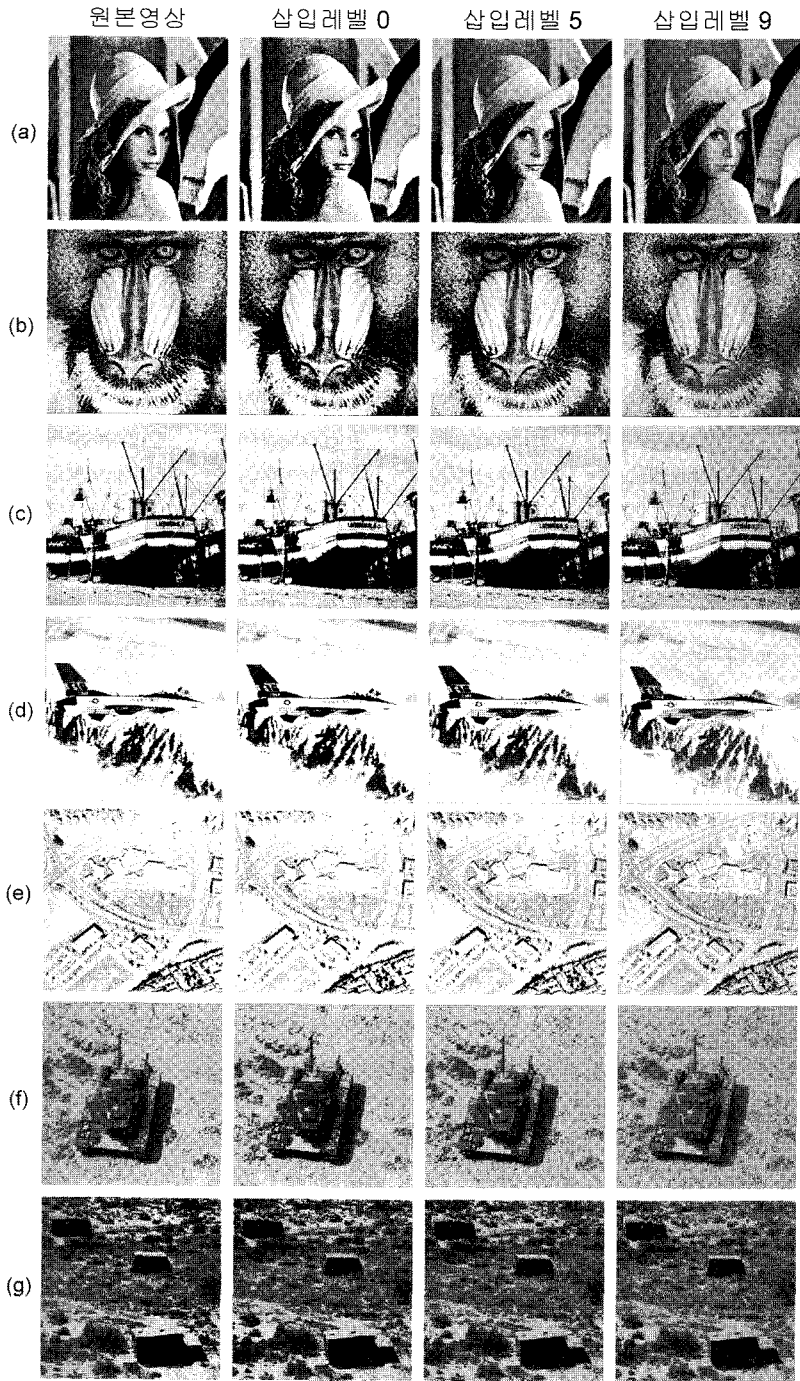


그림 8 삽입레벨에 따른 영상의 변화: (a) Lena, (b) Baboon, (c) Boat, (d) Airplane, (e) Aerial, (f) Tank, (g) Trucks

할 수 없다는 뜻이다. 표에 나열된 알고리즘들의 성능을 보면 워터마크가 삽입된 영상의 품질인 PSNR은 약 48 dB로 대동소이하지만, 유효 삽입용량인 Payload에서 많은 차이를 보인다. 실험 영상에 대하여 제안한 알고리즘

이 기존의 방법보다 10%~344%의 성능 향상을 얻을 수 있었다.

그림 12는 삽입용량에 따른 품질에 대하여 여러 가지 다른 형태의 기존 가역 워터마킹 알고리즘과의 성능 비

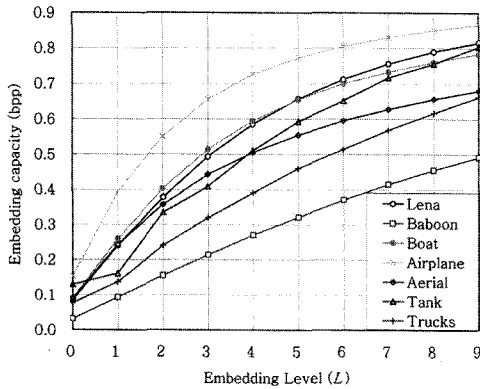


그림 9 삽입레벨에 따른 삽입용량(bpp)

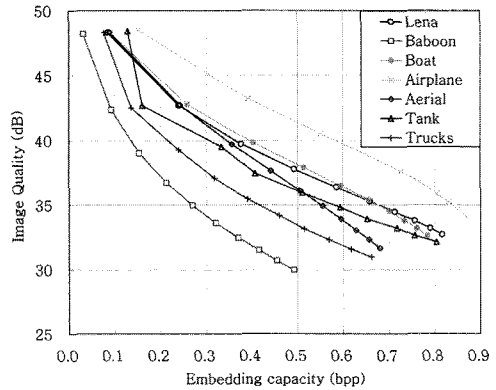


그림 11 삽입용량(bpp)에 따른 품질(dB)

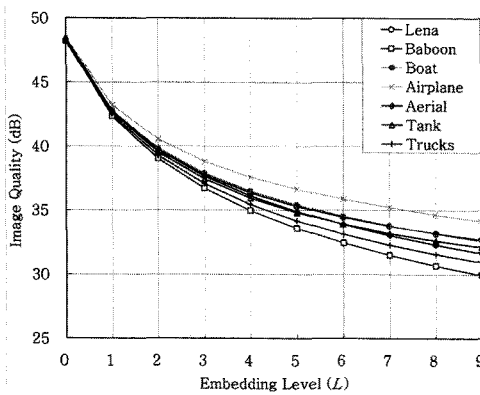


그림 10 삽입레벨에 따른 품질(dB)

교를 보여준다. 성능 비교에 사용된 알고리즘은 2장의 첫 번째 분류에 속하는 영상 압축 방식인 RS 방법[3]과 G-LSB 방법[4], 세 번째 분류에 속하는 차이값 확장 방법인 DE 방법[8], 그리고 네 번째 분류에 속하는 히스토그램 수정 방법인 Ni et al. 방법[12]과 Kim et al. 방법[19]이다. 삽입용량은 오버헤드 정보를 차감한 유효 삽입용량으로 계산되어졌다. DE 방법과 G-LSB 방법이 영상의 품질에 따른 삽입용량이 상대적으로 높게 나타났음에 비하여 RS 방법은 다른 방법들에 비해 낮은 삽

입용량을 가진다. 히스토그램 수정 방법인 Ni et al.의 방법은 48.2 dB의 고정된 품질값을 보이지만 각 영상별로 삽입용량은 거의 차이하지 않는다. 실험 결과들로 볼 때 본 논문에서 제안한 알고리즘이 기존의 다른 가역 워터마킹 방법들에 비하여 낮은 왜곡을 유지하면서 높은 삽입용량을 얻을 수 있음을 확인할 수 있다.

5. 결론

본 논문에서는 영상 콘텐츠에 메시지를 은닉하고, 메시지를 추출한 후에 원본 콘텐츠를 100% 복원하기 위하여, 차이값 히스토그램을 수정하는 가역 영상 워터마킹 알고리즘을 제안하였다. 높은 삽입용량과 지각적 투명성을 얻기 위하여 인접한 픽셀간의 공간적 유사도 특징을 이용하였다. 주요한 기여사항은 오류 예측 보정 기법을 통하여 효과적으로 오버/언더플로우를 방지할 수 있는 방법을 적용하였다. 이를 통하여 기존 오버/언더플로우 해결 방법들이 salt-and-pepper 잡음으로부터 자유롭지 못하다는 한계도 극복하였다. 또한 위치정보 맵을 사용하였음에도 불구하고 실험을 통하여 제안한 알고리즘이 기존의 가역 워터마킹 방법들보다 삽입용량 대비 품질 면에서 좋은 성능을 보임을 증명하였다.

제안한 알고리즘은 의학적/군사적 영상 분석 분야에

표 4 삽입용량 대비 질적품질에 대한 기존 알고리즘과의 성능 비교

	Lena		Baboon		Boat		Airplane	
	Payload	PSNR	Payload	PSNR	Payload	PSNR	Payload	PSNR
Ni et al.'s[12]	5,460	48.2	5,421	48.2	5,394	48.2	16,171	48.2
Varsaki et al.'s[13]	5,460	48.2	5,421	48.2	5,394	48.2	16,171	48.2
Hwang et al.'s[14]	5,408	48.2	5,208	48.2	5,289	48.2	16,119	48.2
Kuo et al.'s[15]	5,418	48.2	5,352	48.2	5,342	48.2	16,125	48.2
Tsai et al.'s[17]	13,459	49.3	N/A	48.1	13,778	49.2	34,171	50.2
Kim et al.'s[19]	20,121	48.9	6,499	48.7	21,442	48.9	32,631	49.0
Proposed	22,220	48.3	8,108	48.2	23,490	48.3	40,025	48.5

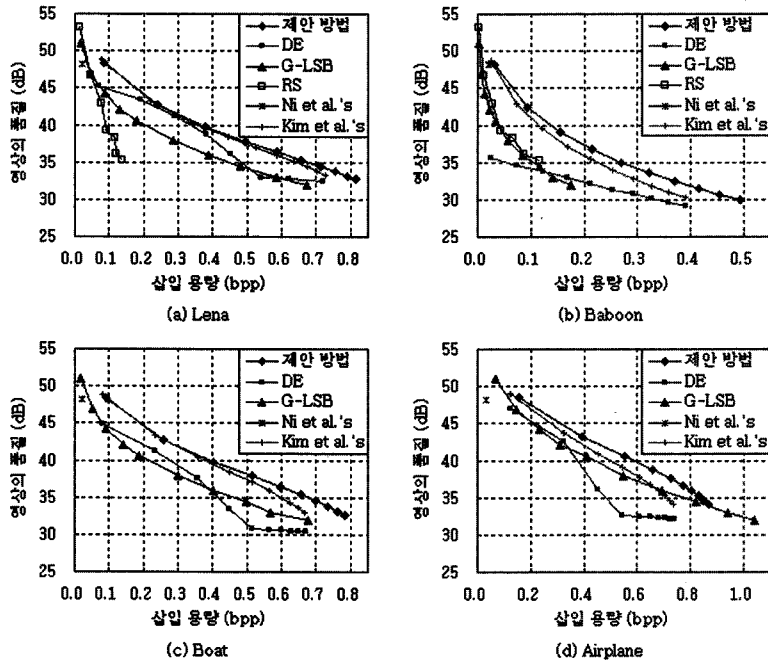


그림 12 삽입용량(bpp)에 따른 품질(dB)에 대한 기존 알고리즘과의 성능 비교

사용되어질 수 있으며, 응용 분야에 요구에 따라 삽입용량을 8k~227k까지로 조절할 수도 있다. 또한 콘텐츠의 인증이나 위조 여부를 확인할 수 있는 연성 워터마킹 방법으로도 사용되어질 수 있다. 본 논문이 극복해야 할 문제는 오버헤드 정보의 양을 더욱 줄이는 것이다. 향후 효과적이면서도 크기가 작은 맵을 구성하여 오버헤드의 정보를 최대한 줄이는 것과, 품질을 좋게 유지하면서도 더 높은 삽입용량을 얻기 위해 성능을 개선할 수 있는 기법에 대하여 연구를 진행할 것이다.

참 고 문 헌

[1] I. J. Cox, M. Miller, J. A. Bloom, J. Fridrich and T. Kalker, *Digital Watermarking and Steganography*, Morgan Kaufmann Publishers Inc., San Francisco, CA, 2007.
 [2] F. Mintzer, J. Lotspiech and N. Morimoto, "Safeguarding digital library contents and users: digital watermarking," *D-Lib Magazine*, 1997.
 [3] J. Fridrich, M. Goljan and R. Du, "Invertible authentication," *Proc. of the SPIE, Security and Watermarking of Multimedia Contents*, San Jose, CA, vol.4314, pp.197-208, 2001.
 [4] M.U. Celik, G. Sharma, A.M. Tekalp and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Trans. on Image Processing*, vol.14, no.2, pp.253-266, 2005.

[5] B. Yang, M. Schmucker, C.B.W. Funk and S. Sun, "Integer DCT-based reversible watermarking for images using compounding technique," *Proc. of the SPIE, Security, Steganography, and Watermarking of Multimedia Contents*, San Jose, CA, vol.5306, pp.405-415, 2004.
 [6] G. Xuan, Q. Yao, C. Yang, J. Gao, P. Chai, Y.Q. Shi and Z. Ni, "Lossless data hiding using histogram shifting method based on integer wavelets," *International Workshop on Digital Watermarking, Lecture Notes in Computer Science*, vol.4283, Springer, Jeju Island, Korea, pp.323-332, 2006.
 [7] S. Lee, C.D. Yoo and T. Kalker, "Reversible image watermarking based on integer-to-integer wavelet transform," *IEEE Trans. on Information Forensics and Security*, vol.2, no.3, pp.321-330, 2007.
 [8] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. on Circuits and Systems for Video Technology*, vol.13, no.8, pp.890-896, 2003.
 [9] A.M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Trans. on Image Processing*, vol.13, no.8, pp.1147-1156, 2004.
 [10] L. Kamstra and H.J.A.M. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting," *IEEE Trans. on Image Processing*, vol.14, no.12, pp.2082-2090, 2005.

[11] D.M. Thodi and J.J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. on Image Processing*, vol.16, no.3, pp.721-730, 2007.

[12] Z. Ni, Y.-Q. Shi, N. Ansari and W. Su, "Reversible data hiding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol.16, no.3, pp.354-362, 2006.

[13] E. Varsaki, V. Fotopoulos and A.N. Skodras, "A reversible data hiding technique embedding in the image histogram," *Technical Report HOU-CS-TR-2006-08-GR*, Hellenic Open University, 2006.

[14] J.H. Hwang, J.W. Kim and J.U. Choi, "A reversible watermarking based on histogram shifting," *International Workshop on Digital Watermarking, Lecture Notes in Computer Science*, vol.4283, Springer-Verlag, Jeju Island, Korea, pp.348-361, 2006.

[15] W.-C. Kuo, D.-J. Jiang and Y.-C. Huang, "Reversible data hiding based on histogram," *International Conference on Intelligent Computing, Lecture Notes in Artificial Intelligence*, vol.4682, Springer-Verlag, Qing Dao, China, pp.1152-1161, 2007.

[16] C.-C. Lin, W.-L. Tai and C.-C. Chang, "Multi-level reversible data hiding based on histogram modification of difference images," *Pattern Recognition*, vol.41, no.12, pp.3582-3591, 2008.

[17] P. Tsai, Y.-C. Hu and H.-L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Processing*, vol.89, no.6, pp.1129-1143, 2009.

[18] S.-K. Lee, H.-M. Yoo, Y.-H. Suh and J.-W. Suh, "Improved Reversible Data Hiding Based on Histogram Modification of Difference Images," *International Conference on Consumer Electronics*, pp.181-182, 2010.

[19] K.-S. Kim, M.-J. Lee, H.-Y. Lee and H.-K. Lee, "Reversible data hiding exploiting spatial correlation between sub-sampled images," *Pattern Recognition*, vol.42, no.11, pp.3083-3096, 2009.

[20] C. W. Honsinger, P. Jones, M. Rabhani and J. C. Stoffel, "Lossless recovery of an original image containing embedded data," *US Patent application*, Docket No: 77102/E-D, 1999.



이 해 연

1997년 성균관대학교 정보공학과(학사)
1999년 한국과학기술원 전산학과(공학석사)
2006년 한국과학기술원 전자전산학과 전산학전공(공학박사). 2001년~2006년 (주)세트렉아이 선임연구원. 2006년~2007년 코넬대학교 박사후연구원. 2008년~현재 국립금오공과대학교 컴퓨터공학부 교수. 관심분야는 멀티미디어, 영상처리, 콘텐츠보안, 디지털워터마킹 등



김 병 만

1987년 서울대학교 컴퓨터공학과(학사)
1989년 한국과학기술원 전산학과(석사)
1992년 한국과학기술원 전산학과(박사)
1992년~현재 금오공과대학교 교수.
1998년~1999년 미국 UC, Irvine 대학 방문교수. 2005년~2006년 미국 콜로라도 주립대학 방문교수. 관심분야는 인공지능, 정보검색, 정보보안



김 경 수

2005년 인하대학교 컴퓨터공학과(학사)
2007년 한국과학기술원 전산학과(공학석사). 2010년 한국과학기술원 전자전산학과 전산학전공(공학박사). 2010년~현재 KT 네트워크연구소 재직. 관심분야는 멀티미디어, 워터마킹, 디지털 포렌식, 네트워크 보안 등



여 동 규

1999년 금오공과대학교 컴퓨터공학과(학사). 2001년 금오공과대학교 컴퓨터공학과(석사). 2003년 금오공과대학교 컴퓨터공학과(박사 수료). 관심분야는 정보보호, 디지털 워터마킹, 디지털 포렌식 등