

스마트폰에서 디스플레이-캡처 영상의 왜곡 분석에 기반한 모바일 워터마킹

배종욱[†], 정성환^{**}

요 약

본 논문은 스마트폰에서 디스플레이-캡처 영상의 왜곡 분석에 기반한 모바일 워터마킹 기법을 제안한다. 제안한 기법에서 워터마크 삽입을 위해 웨이블릿 변환 영역에서 주파수 대역의 특성을 이용하여 랜덤시퀀스를 구성한다. 웨이블릿 변환 후 선택된 부대역의 각 블록 계수와 랜덤시퀀스를 계수비교합산법(Coefficients Comparative Sum, CCS)으로 계산하고, 삽입 임계값을 이용하여 반복적으로 워터마크를 삽입하여 워터마크 강도를 높인다. 디스플레이-캡처 과정에서 발생하는 왜곡을 보정하기 위해 영상 외부에 프레임을 추가하며, 이를 검출함으로써 워터마크 동기를 맞춘다. 또한 프레임 임계값 설정을 위해 반복적 적응 임계값을 사용하여 프레임 검출율을 향상시킬 수 있다. 제안한 방법은 256×256 크기의 표준 디지털 영상에 206 bits의 정보를 삽입하여 PSNR이 평균 약 41.42 dB를 나타내었다. 또한 디스플레이-캡처 영상의 워터마크 추출 실험에서, 캡처된 영상의 97% 이상에서 프레임 교차점을 정확히 인식하였으며, 비트에러율이 평균 약 3.73% 을 기록하여, 기존 Pramila 방법에 비해 비트에러율이 70% 이상 향상되었다.

Mobile Watermarking Based on the Distortion Analysis of Display-Capture Image in a Smart Phone

Jong-Wook Bae[†], Sung-Hwan Jung^{**}

ABSTRACT

In this paper, we propose a mobile watermarking based on the distortion analysis of display-capture image in a smart phone. We compose a random sequence by utilizing the property of frequency band in the wavelet domain. Then we calculate the CCS (Coefficients Comparative Sum) using the block wavelet coefficients of selected subbands after the wavelet transformation and the random sequence and repeatedly embed a watermark using an insertion threshold for the watermark robustness. For correcting a distortion caused by the display-capture process, we adopt a frame at the outside of watermarked image, then we can equate a watermark synchronization by detecting the frame. And we can improve frame detection ratio by using an iteratively adaptive threshold. A proposed scheme embedded information of 206 bits into standard digital images and it shows an average about 41.42 dB in PSNR. In watermark extract experiments, a proposed scheme accurately recognizes the frame more than 97% in total captured images. Also in BER (Bit Error Ratio) of captured images, it shows about 3.73%, then it was improved more than 70%, compared with the Pramila's method.

Key words: mobile(모바일), display-capture(디스플레이-캡처), smart phone(스마트폰), digital watermarking(디지털 워터마킹)

※ 교신저자(Corresponding Author) : 정성환, 주소 : 경상남도 창원시 의창구 사림동 9번지 창원대학교 55호관 505호실(641-773), 전화 : 055)213-3823, FAX : 055)286-7429, E-mail : sjung@changwon.ac.kr
접수일 : 2012년 3월 15일, 수정일 : 2012년 5월 31일

완료일 : 2012년 6월 4일

[†] 준회원, 창원대학교 컴퓨터공학과
(E-mail : bae@cwnu.ac.kr)

^{**} 중신회원, 창원대학교 컴퓨터공학과

1. 서 론

디지털 워터마킹 기술의 출현과 발전으로 워터마킹을 적용할 수 있는 다양한 미디어에 대한 연구가 이루어지고 있다. 초기의 워터마킹 기술들은 주로 영상이나 오디오 같은 디지털 미디어에 워터마크를 삽입하는 연구가 대부분 이었다[1]. 그 후로 인쇄된 미디어에서 워터마크를 검출하려는 요구로, 프린트-스캔(print-scan) 처리에 강인한 워터마킹 방법에 대한 연구가 이루어졌다[2-4]. 또한 최근 컴퓨터, 전화, 보안 등의 영역에서 웹 카메라의 사용이 증가하고 있으며, 이에 따라 프린트-캠(print-cam) 처리에 강인한 워터마킹 방법에 대한 연구가 이루어지고 있다[5-7].

한편, 이동통신 기술의 발전으로 인해 모바일 환경에 적용하기 위한 다양한 서비스들이 개발되고 있다[8]. 그로 인하여 모바일 환경에 적합한 디지털 워터마킹 기술의 적용이 필요하다. 예로서, 모바일 환경에서 디지털 영상이 LCD(Liquid Crystal Display) 모니터에 디스플레이되어 활용되는 상황을 생각할 수 있다. 이동 중에 어디선가 영상이 불법적으로 사용되어 디스플레이되어도 현재로서는 그 대책이 없는 실정이다. 따라서 LCD 모니터에 디스플레이된 영상의 합법적인 사용 여부를 즉각적으로 확인할 수 있는 방법이 필요하다. 이러한 디스플레이-캡처 환경에서는 캡처된 영상 그 자체로 다양한 왜곡을 포함하고 있으며, 이 왜곡들은 영상에서 워터마크의 추출을 방해하는 요소로 작용한다. 여기서 필요한 것이 디스플레이-캡처(display-capture) 과정의 왜곡 분석에 근거한 워터마킹의 적용이다.

기존 논문에서는 모바일 환경에서 워터마크 기술을 적용한 예를 찾아보기 힘들다. 다만, Digimarc 사는 프린트-캠 처리에서 워터마크를 추출하기 위해 발생하는 다양한 왜곡을 기술하였다[5]. 또한 Nakamura, Pramila 등은 프린트-캠 처리로 인한 왜곡을 해결하는 방법으로서 워터마크된 영상의 외부에 프레임을 추가하고, 이 프레임을 검출함으로써 왜곡을 보정하는 방법을 제안하였다[6,7]. 그러나 이 논문들은 LCD 모니터에 디스플레이된 영상이 아닌 종이에 인쇄된 영상을 대상으로 한 것이다. 또한 다양한 왜곡의 해결 방법과 프레임의 검출 방법에 대해서는 자세하게 기술하지 않았다.

따라서 본 논문은 모바일 환경에서 디스플레이-

캡처 영상의 왜곡 분석을 기반으로 정보 보안을 위한 모바일 워터마킹(mobile watermarking) 기법을 제안한다. 먼저, 디지털 영상을 LCD 모니터에 디스플레이 할 때의 특성과 스마트폰의 카메라로 캡처하는 과정에서 발생하는 특성을 분석하여, 어떠한 특성들이 워터마크의 추출을 방해하고 워터마크의 생존율을 감소시키는지 기술한다. 그리고 이러한 분석을 바탕으로 시각적으로 덜 민감한 색상으로 프레임 색상을 설정하여 워터마크가 된 영상의 외부에 추가하고, 캡처 후에 영상의 다양한 왜곡에도 프레임을 정확히 검출할 수 있는 기법을 제시한다. 또한 화질을 손상시키지 않으면서 워터마크의 강인성을 높이기 위한 효과적인 방법들을 제안한다.

본 논문의 나머지 부분은 다음과 같이 구성된다. 2장에서는 디스플레이-캡처 처리의 의미와 이러한 과정에서 발생하는 왜곡의 종류를 기술한다. 3장에서는 이러한 왜곡에도 워터마크를 생존율을 높이기 위해 웨이브릿 기반의 삽입 방법과 왜곡보정을 위한 프레임 검출 및 워터마크 디코딩 방법을 제시한다. 4장에서는 워터마크 추출 성능을 개선하기 위한 방법을 기술한다. 5장에서는 스마트폰 상에서 다양한 실험을 통해서 최적의 방법을 찾고 기존 논문의 방법과 비교 실험하고, 6장에서 결론을 맺는다.

2. 모바일 디스플레이-캡처 환경의 왜곡 분석

2.1 모바일 디스플레이-캡처 시나리오

디지털 영상을 인간이 시각적으로 인지하는 방법으로 일반적으로 프린터를 이용한 인쇄의 방법과 모니터에 디스플레이하는 방법이 있다. 또한 아날로그화된 영상은 다양한 용도에 활용하기 위해 다시 디지털화될 수 있다. 그 방법으로는 스캐너에 의한 스캐닝과 디지털 카메라에 의한 캡처가 있다. 스캐닝의 방법은 고정된 장소에서 고해상도 영상을 획득하는데 장점이 있지만 모바일 환경에서는 디지털 카메라에 비해 그 활용에 제약이 있다. 한편, 스마트폰의 대중화와 함께 모바일 환경에서 아날로그 영상을 디지털화하여 이용하는 방법으로 스마트폰 카메라가 새로운 대안이 되고 있다.

모바일 디스플레이-캡처 과정에서는 다양한 왜곡이 발생할 수 있다. 그림 1에서 워터마크의 추출을 방해할 수 있는 왜곡들을 도식화 하고 식 (1)과 같이

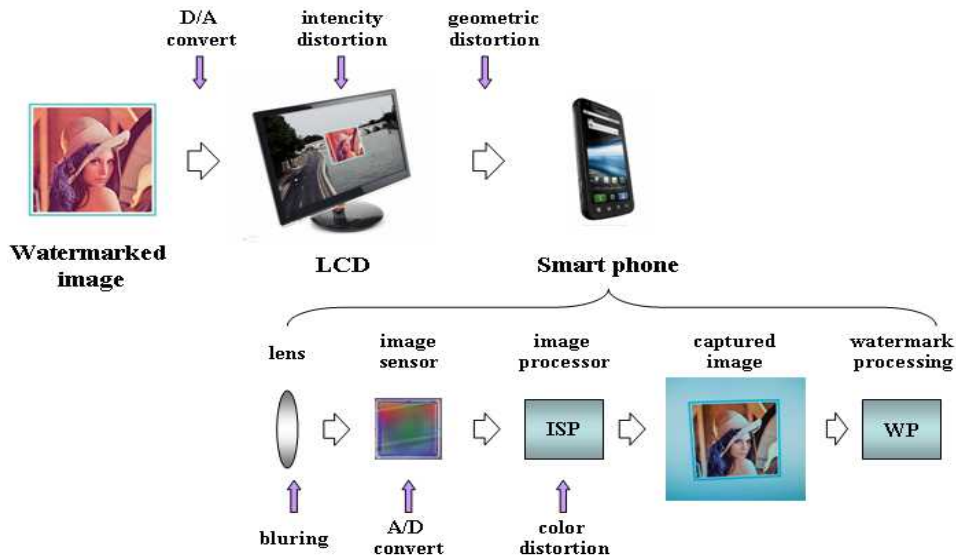


그림 1. 디스플레이-캡처 과정의 왜곡

표현할 수 있다.

$$I_{analog} = LCD_D(I_{digital}) \tag{1}$$

$$I_{digital} = ISP_D(Sensor_D(Lens_D(I_{analog})))$$

디지털 영상이 LCD에 의해 아날로그 영상이 되며, 이 과정에 D/A 컨버팅, 밝기왜곡 그리고 사용자가 캡처하는 과정에서 기하학 왜곡 등이 발생할 수 있다. 그리고 스마트폰 카메라에 의해 디지털화 되는 과정에서 렌즈, 이미지 센서, ISP (Image Signal Processor) 등에 의해 왜곡이 발생할 수 있다.

2.2 디지털 컨버팅에 의한 왜곡

모바일 디스플레이-캡처는 디지털 영상을 LCD에 디스플레이하고, 스마트폰의 카메라에 의해 캡처가 이루어진다. 이 과정에서 디지털 영상이 아날로그화되며, 다시 아날로그 영상은 스마트폰의 카메라에 의해 디지털화 된다. 따라서 D/A 컨버전과 A/D 컨버전이 동시에 발생하며, color interpolation, white balance, 압축 등이 추가되어 화소값에서 상당한 열화가 발생한다[5]. 따라서 이러한 화소값의 열화에도 워터마크가 살아남을 수 있는 워터마크 삽입 도메인과 삽입 방법의 선택이 중요하다.

2.3 캡처 동작에 의한 왜곡

디지털 카메라로 캡처하여 획득된 영상은 원본 디지털 영상에서 외부 배경을 포함하게 되며, 회전, 크

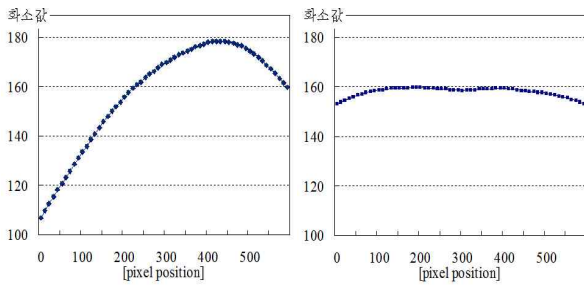
기변환, 평행이동과 같은 기하학적 왜곡이 발생하게 된다. 또한 피사 영상의 평면과 카메라 렌즈의 축이 수직하지 않으면 투영 왜곡도 발생하게 된다[9]. 이러한 왜곡은 디지털 영상에 삽입된 워터마크의 동기를 잃어버리게 함으로써 워터마크의 검출에 상당한 어려움을 초래한다. 기하학적 왜곡을 보정하기 위한 기존의 방법들이 존재하지만, 투영 왜곡은 보정하는데 많은 어려움이 있다.

2.4 카메라 렌즈에 의한 왜곡

스마트폰 앱은 캡처 과정에서 영상을 정확하게 획득하기 위해서 주기적으로 렌즈의 초점을 조절하도록 설계해야 한다. 이로 인해 특정 캡처 영상은 오토포커싱을 하는 과정에서 초점의 불일치로 인해 블러링이 심하게 발생할 수 있다. 또한 이러한 블러링은 캡처 과정에서 스마트폰의 흔들림에 의해서도 발생할 수 있다. 이렇게 블러링이 심한 영상은 프레임 화소의 오인식이 증가할 뿐 아니라 정확히 검출된 프레임 영상에서도 워터마크의 생존율이 현저히 떨어질 수 있다.

2.5 LCD 백라이트에 의한 왜곡

LCD는 각각의 컬러 화소를 백라이트의 밝기에 의해 색상을 표현한다. 그러나 LCD 모니터의 휘도는 한 모니터 내에서 같은 값을 표현하더라도 다른 휘도를 가질 수 있다[10]. 그림 2는 모니터에 단일 색상



(a) 세로 방향 (b) 가로 방향

그림 2. LCD 모니터의 부분별 밝기 비교

을 디스플레이하고 이 영상을 스마트폰으로 캡처하여 영상의 좌표 위치에 따른 LCD의 휘도를 실험한 것이다.

그림 2 (a)는 세로 방향, (b)는 가로 방향 휘도 평균을 나타낸 것이다. 그림 2(a)에서, 그래프의 왼쪽, 모니터에서는 상단이 가장 어두운 휘도를 나타내며, 내려갈수록 밝아지다가 모니터의 하단에서 다시 조금 어두워지는 경향을 나타낸다. 그림 2(b)에서 모니터의 좌우로는 중간 부분에서 대부분 일정한 휘도를 나타내고, 가장자리에서 조금 낮은 휘도를 나타낸다. 따라서 캡처 영상의 가장자리 부분을 제외하고 중심 부분만을 이용하여 워터마크 추출을 수행하는 것이 바람직하다. 또한 모니터의 세로 방향으로 휘도의 편차가 크기 때문에 프레임 화소를 인식하기 위한 임계값을 설정하는데 어려움이 있고, 이로 인해 그림 3과 같이 프레임 화소를 완전히 검출하지 못하는 경우가 발생할 수 있다.

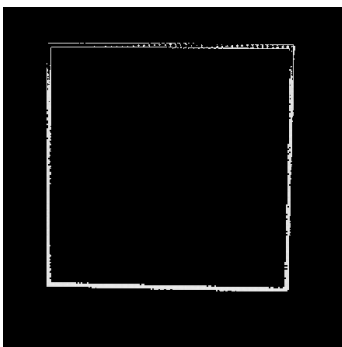
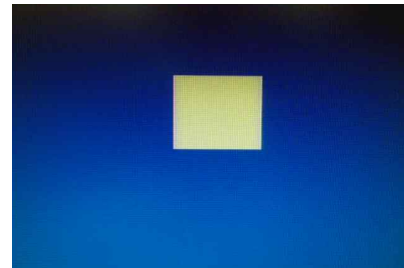


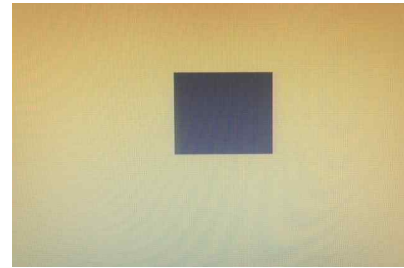
그림 3. 프레임 오검출 영상

2.6 카메라 센서 및 영상처리에 의한 왜곡

그림 4는 특정 색상을 카메라 센서가 어떻게 인식하는지를 알아보기 위한 실험이다. 스마트폰의 경우



(a) 내부 YUV (120 125 130)
외부 YUV (74 177 86)



(b) 내부 YUV (120 125 130)
외부 YUV (182 82 169)

그림 4. 대비 색상 캡처 영상

카메라 센서에 의해 획득된 데이터는 RGB 컬러 공간이 아닌 YUV 컬러 공간을 사용한다. 따라서 본 실험은 YUV 채널을 이용하여 특정 값으로 영상을 생성하고, LCD에 디스플레이 한 후 캡처한다.

두 영상 모두 내부의 색상은 YUV 채널로 120, 125, 130의 값으로 동일한 값을 설정하고, 외부의 색상은 그림 4(a)의 경우 74 177 86 으로 (b)의 경우 182 82 169로 설정하여 각 채널의 값이 중간 값인 내부 색상과 대비될 수 있도록 하였다. 즉, Y 채널은 내부 색상 120을 기준으로 하여, 외부 색상을 (a)는 74, (b)는 182 으로 설정하였다. U 채널은 내부 색상 125를 기준으로 외부 색상을 (a)는 177, (b)는 82 으로 설정하며, V 채널은 내부 색상 130을 기준으로 (a)는 86, (b)는 169를 설정하였다.

색상이 어떻게 인식되는지 정확히 확인하기 위해서 두 영상을 그림 5~7과 같이 각 채널별로 히스토그램으로 표현한다. 모든 채널에서 막대그래프는 그림 4(a)의 히스토그램을, 꺾은선 그래프는 그림 4(b)의 히스토그램을 나타낸다. 먼저 그림 5의 Y 채널의 경우, 막대그래프의 히스토그램에서 오른쪽의 150~210 정도가 내부 색상을 나타내며, 꺾은선 그래프의 히스토그램에서는 왼쪽의 70~110 정도가 내부 색상을 나타낸다. 또한 그림 6의 U 채널의 경우, 막대그래프의 히스토그램에서 왼쪽의 80~110 정도가 내부

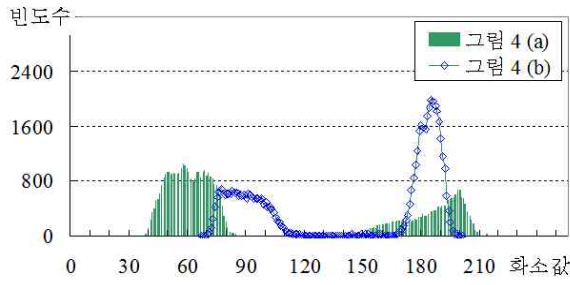


그림 5. Y 채널 히스토그램

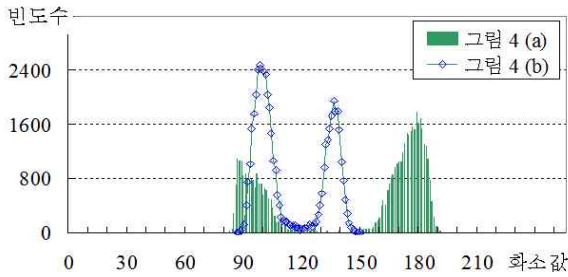


그림 6. U 채널 히스토그램

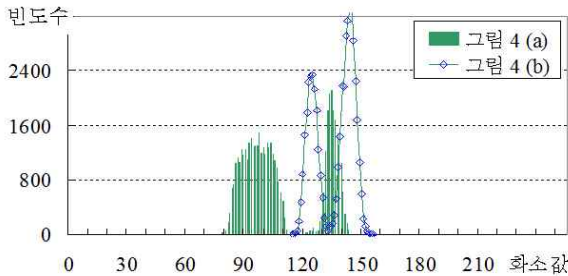


그림 7. V 채널 히스토그램

색상을 나타내며, 꺾은선 그래프의 히스토그램에서는 왼쪽의 130~140 정도가 내부 색상을 나타낸다. 그림 7의 V 채널의 경우, 막대그래프의 히스토그램에서 왼쪽의 130~140 정도가 내부 색상을 나타내며, 꺾은선 그래프의 히스토그램에서는 왼쪽의 120~130 정도가 내부 색상을 나타낸다.

그림 5~7을 통해서, YUV 채널로 120, 125, 130을 나타내는 내부 색상은 외부 색상에 따라서 서로 다른 화소값으로 캡처됨을 알 수 있다. 이것은 디지털 카메라가 정확한 화소값을 표현하는 것이 아니라 인간 시각 시스템과 최대한 유사하게 인식하도록 설계되기 때문에 발생하는 현상이다[5]. 이러한 문제는 캡처 영상 전체에서 일률적으로 나타나기 때문에 삽입된 워터마크의 훼손은 심하지 않을 수 있다. 그러나 워터마크 동기를 맞추기 위한 프레임 검출 과정에서 오인식이 발생하여 워터마크를 검출할 수 없게 된다.

3. 웨이브릿 기반의 모바일 워터마킹

3.1 워터마크 삽입 알고리즘

워터마크 삽입은 먼저 원본 영상을 컬러 공간 변환하여 워터마크를 삽입할 명암도 영상을 분리한다. 스마트폰의 카메라가 YUV 컬러 공간을 사용하여 화소를 획득하기 때문에 검출 과정에서의 속도와 성능 향상을 위해 명암도 영상은 YUV 컬러 공간에서 Y 채널을 선택한다.

다음으로 웨이브릿 변환을 이용하여 명암도 영상을 부대역으로 분리한다. 각 대역마다 계수들은 특유의 성질을 갖는다. 즉, LL 대역은 원영상의 통계적 성질을 유지하며, LH 대역은 수직 방향으로 고주파 성분을 가지며, HL 대역은 수평 방향으로 고주파 성분을 가진다.

선택된 삽입 도메인은 식 (2)와 같이 $N \times M$ 크기의 겹치지 않는 블록으로 나눈다. 여기서, B_k 는 k 번째 $N \times M$ 크기 블록이고 K 는 블록의 총 개수이다. 각 블록에 한 비트의 정보를 삽입한다.

$$I(x, y) = \bigcup_{k=0}^{K-1} B_k, \bigcap_{k=0}^{K-1} B_k = \phi, \quad k = \{0, 1, 2, \dots, K-1\} \quad (2)$$

워터마크 삽입의 과정은 먼저 각 블록에 식 (3)의 수식으로 계수비교합(Coefficients Comparative Sum, CCS)을 구한다. 여기서 $B_k(n, m)$ 는 k 번째 블록의 웨이브릿 계수들이며, $S(n, m)$ 는 1, -1의 값을 갖는 랜덤 시퀀스로 워터마크 추출 시 비밀키의 역할을 한다. 또한 이 과정에서 웨이브릿 변환 계수의 대역별 특성을 이용하여 워터마크의 검출 성능을 향상시킬 수 있다. 즉, 웨이브릿 부대역 중 LH 대역에 삽입 시에는 랜덤 시퀀스 $S(n, m)$ 를 수직 방향으로 짝을 이루어 1, -1이 배치되게 하며, HL 대역에 삽입 시에는 수평 방향으로 1, -1이 배치되게 구성한다. 본 논문은 LH 대역에 대해서만 삽입하여 실험한다.

$$CCS_k = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} B_k(n, m) * S(n, m) \quad (3)$$

디스플레이-캡처 과정에서도 워터마크가 살아남을 수 있도록 강인한 삽입을 위해서 하나의 블록 내에서 식 (4)를 만족할 때까지 식 (5)를 반복적으로 수행한다. 여기서 W_k 는 워터마크 비트에 에러 정정

코드를 적용한 시퀀스로 1, -1의 값을 갖고, 삽입 임계값(Th_{insert})과 α 는 워터마크의 삽입 강도를 조절하는 값이다.

$$CCS_k * W_k > Th_{insert} \tag{4}$$

$$B_k(n,m)' = B_k(n,m) + \alpha * S(n,m) * W_k \tag{5}$$

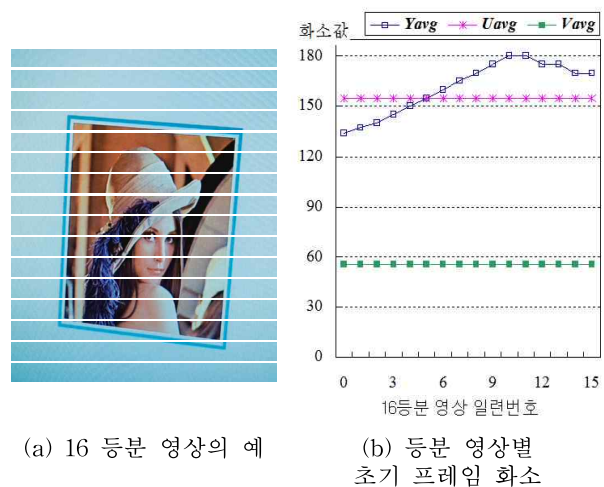
모든 블록에 대해 워터마크 삽입이 완료되면, 역 웨이브릿 변환을 수행하여 공간 도메인으로 환원한다. 그리고 워터마크된 명암도 영상과 남겨진 U, V 채널을 RGB 컬러 공간으로 복원한다. 마지막으로 워터마크된 영상의 외부에 특정색의 프레임을 추가한다. 이것은 워터마크 추출 과정에서 캡처된 영상의 기하학 왜곡 및 투영 왜곡을 보정하기 위한 것이다.

3.2 워터마크 추출 알고리즘

3.2.1 프레임 임계값 산정

디스플레이-캡처 과정에서 발생하는 다양한 왜곡을 2장에서 제시하였으며, 일반적으로 LCD는 같은 색상이라도 세로방향으로 휘도의 차이가 심함을 그림 2 (a)와 그림 3을 통해 알 수 있었다. 이러한 휘도의 차이에도 불구하고 프레임 영상을 정확히 검출하기 위해서, 캡처 영상 내에서 프레임 임계값을 다르게 적용할 수 있다. 그 방법으로 캡처 영상을 16 등분하여 각 부분영상별로 프레임 화소값을 설정한다. 실험에 사용된 많은 캡처 영상의 평균적인 값을 이용하여 YUV에 해당하는 초기 프레임 화소값을 설정한다. 그림 8 (a)는 캡처 영상을 16 등분한 예를 나타낸 것이며, (b)는 초기 프레임 화소값을 각 분리 영상별로 표시한 것이다.

다음으로 프레임 임계값은 초기 프레임 화소에 채



(a) 16 등분 영상의 예

(b) 등분 영상별 초기 프레임 화소

그림 8. 캡처 영상의 분리 및 등분 영상별 초기 프레임 화소값

널별 간격($\delta_Y, \delta_U, \delta_V$)을 식 (6)과 같이 적용하여 산출된다. 이 임계값은 범위를 갖는 값으로 각각 16 등분의 부분 영상에 해당하는 값을 갖는다. 여기서 $\delta_Y, \delta_U, \delta_V$ 는 실험적으로 결정된다.

$$\begin{aligned} Th_{Y_{begin}(i)} &= Y_{avg(i)} - \delta_Y, & Th_{Y_{end}(i)} &= Y_{avg(i)} + \delta_Y \\ Th_{U_{begin}(i)} &= U_{avg(i)} - \delta_U, & Th_{U_{end}(i)} &= U_{avg(i)} + \delta_U \\ Th_{V_{begin}(i)} &= V_{avg(i)} - \delta_V, & Th_{V_{end}(i)} &= V_{avg(i)} + \delta_V \end{aligned} \tag{6}$$

$i = \{0,1,2,\dots,15\}$

3.2.2 프레임 영상 검출

계산된 임계값으로 식 (7)을 영상 내의 모든 화소에 적용하여 그림 9와 같이 채널별 프레임 영상을 검출한다. 그리고 프레임 직선 검출을 위해서는 3개의 프레임 영상 중 어느 것을 사용해도 무방하나, 상대적으로 큰 값을 갖고 16 등분 부분 영상에서 일정한 값을 유지하는 U_{frame} 을 최종 프레임 영상으로 선택한다.

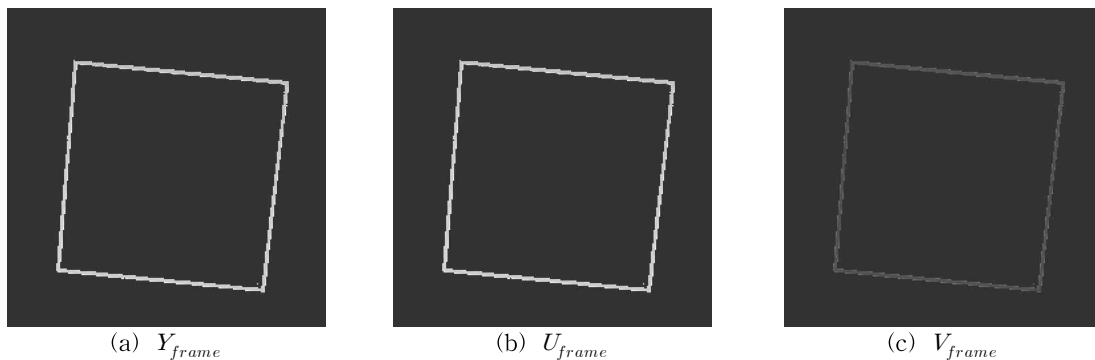


그림 9. 채널별 초기 프레임 영상

$$\begin{aligned}
 & \text{if } (Th_{V_begin(i)} < Y(x,y) < Th_{V_end(i)} \ \& \\
 & \quad Th_{U_begin(i)} < U(x,y) < Th_{U_end(i)} \ \& \\
 & \quad Th_{V_begin(i)} < V(x,y) < Th_{V_end(i)}) \\
 & \text{then } Y_{frame}(x,y) = Y(x,y), U_{frame}(x,y) = U(x,y), \\
 & \quad V_{frame}(x,y) = V(x,y) \\
 & \text{else } Y_{frame}(x,y) = 0, \quad U_{frame}(x,y) = 0, \\
 & \quad V_{frame}(x,y) = 0
 \end{aligned} \tag{7}$$

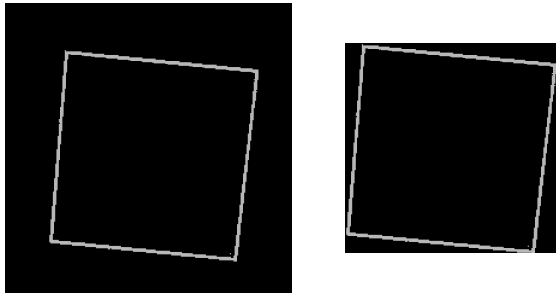


그림 10. 외부 배경이 제거된 프레임 영상의 예

3.2.3 프레임 외부 배경 제거

정상적으로 캡처된 영상은 워터마크가 삽입된 영상부와 영상을 둘러싼 프레임 그리고 외부 배경을 포함하고 있다. 이 프레임을 이용하여 그림 10과 같이 프레임 외곽 영역을 제거함으로써 이후 프레임 직선 검출에서 계산량을 줄일 수 있으며, 직선의 오인식으로 인한 에러 가능성도 감소할 수 있다. 먼저, 프레임 영상을 가로와 세로 방향으로 각각 화소합을 구하여 그림 11과 같이 그래프로 나타낸다. 그리고 그래프의 가장자리부분에서 최대값의 위치를 찾아 프레임 상하좌우 외곽 위치를 추정할 수 있다.

3.2.4 워터마크 영상 분리

외부 배경이 제거되었음에도 캡처 영상에는 아직

회전, 평행이동, 크기 변화와 같은 다양한 기하학적 왜곡 및 투영 왜곡 등이 남아있다. 여기서 워터마크를 정상적으로 추출하기 위해서는 이러한 왜곡을 보정하여 워터마크 삽입 시와 같게 워터마크 동기를 일치시켜야 한다. 이를 위해 워터마크 삽입 과정에서 워터마크된 영상 외부에 특정색의 프레임을 추가하였고, 이 프레임의 직선을 이용하여 4개의 교차점 좌표를 구할 수 있다. 다음으로 식 (8)을 적용하여 왜곡 계수를 산출하여 왜곡 영상을 보정하면, 실제 워터마크가 삽입된 부분만을 분리할 수 있다[11-12].

$$(x,y) = \left(\frac{p_1 X + p_2 Y + p_3}{p_7 X + p_8 Y + p_9}, \frac{p_4 X + p_5 Y + p_6}{p_7 X + p_8 Y + p_9} \right) \tag{8}$$

여기서 x,y 는 원 영상의 프레임 모서리 좌표이고, X,Y 는 캡처된 영상의 프레임 모서리 좌표이며, $p_1 \sim p_9$ 는 보정 계수이다.

3.2.5 워터마크 디코딩

워터마크 영상을 분리한 다음으로 삽입 정보의 디코딩을 위해서는 워터마크 삽입에 사용했던 도메인으로 변경해야한다. 즉, 분리된 영상을 웨이브릿 변환하고 LH 혹은 HL 밴드가 워터마크 추출의 대상 도메인으로 선택된다. 선택된 도메인에 대해 워터마크 삽입 시와 같이 각 블록으로 분리하고 식 (3)의 CCS_k 를 계산한 후, 식 (9)의 조건에 따라 각 블록의 삽입 정보(W_k)를 결정한다.

$$W_k = \begin{cases} +1 & \text{if } CCS_k \geq 0 \\ -1 & \text{if } CCS_k < 0 \end{cases} \tag{9}$$

모든 블록의 계산이 끝나면 마지막으로 추출된 신호에 생길 수 있는 오류 비트를 정정하기 위해 ECC

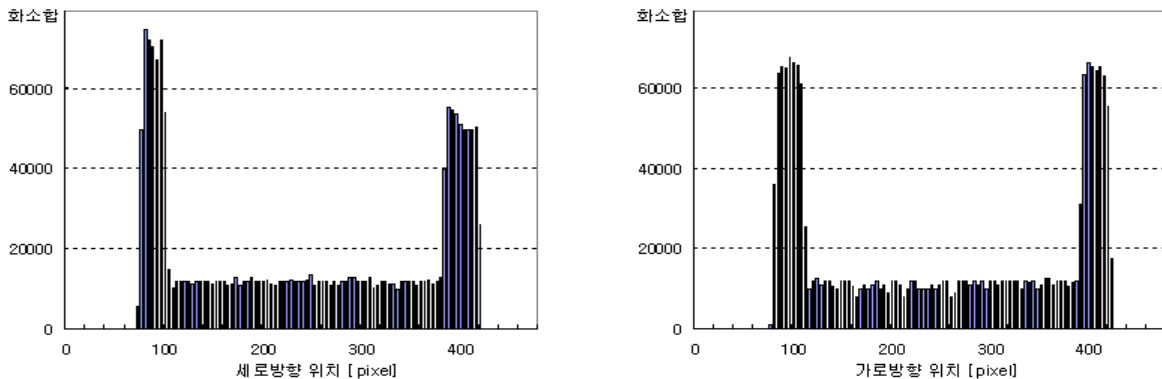


그림 11 프레임 영상의 방향별 화소합

디코딩을 수행할 수 있다.

4 워터마크 추출 성능 개선

4.1 프레임 오인식 영상 배제

프레임 영상에서 교차점 좌표에 대한 오인식은 바로 워터마크 추출의 오류로 나타난다. 따라서 교차점 오인식이 발생한 캡처 영상은 워터마크 디코딩 이전에 제거하고, 다음 영상을 입력 받는 것이 바람직하다. 이것은 스마트폰이 초당 최대 30개의 영상을 연속적으로 획득할 수 있기 때문에 쉽게 적용할 수 있다.

프레임 검출과 워터마크 영상 분리 과정에서 구해진 4개의 교차점 좌표를 이용해 프레임 사각형의 각 선분의 길이를 구할 수 있다. 본 연구에서는 프레임의 각 선분들의 차분을 이용하여 배제 영상을 정한다. 즉, 식 (10)과 같이 선분 길이 평균에 대한 차분의 비율 D(%)를 산출한다. 이 값은 분리된 워터마크 영상의 투영 왜곡의 정도를 나타내는 것이다. 영상 캡처의 미숙이나 캡처 영상의 블러링 등으로 인해 프레임 직선이 오인식되는 경우, 이 값은 상대적으로 큰 값을 갖는다. 본 실험에서는 D가 4% 이상인 경우 해당 영상을 디코딩 과정에서 배제한다[10].

$$\begin{aligned}
 Linc_i &= \sqrt{(x_i + x_{i\oplus 1})^2 + (y_i + y_{i\oplus 1})^2}, < i \oplus 1: (i+1) \bmod 4 > \\
 Aver_{Linc} &= \frac{1}{4} \sum_{i=0}^3 Linc_i \\
 Aver_{diff} &= \frac{1}{6} \sum_{i=0}^2 \sum_{j=i+1}^3 |Linc_i - Linc_j| \\
 D(\%) &= \frac{Aver_{diff}}{Aver_{Linc}} * 100
 \end{aligned}
 \tag{10}$$

4.2 블러링 영상의 배제

캡처 영상에서 블러링이 심한 경우 프레임 화소의 오인식이 발생할 수 있다. 프레임 오인식은 그 차제로 워터마크 검출이 불가능하다. 뿐만 아니라 블러링으로 인한 화소의 열화는 워터마크 생존율을 급격히 감소시킨다. 따라서 심한 블러링이 발생한 영상은 워터마크 디코딩 과정에서 미리 배제하는 것이 바람직하다.

블러링의 정도를 분석하는 다양한 방법이 있지만, 본 연구가 스마트폰이라는 환경상의 제약과 더불어 실시간 처리를 수행해야하는 제약이 있다. 이러한 제약으로 인해 수행시간을 감소시킬 수 있는 방법으로 써, 식 (11)과 같이 이웃 화소간 차분이 특정값 이상

인 값들의 평균을 이용하여 주파수 성분을 분석한다. 여기서 $p(x,y)$ 는 캡처 영상에서 x, y 좌표의 화소값이고, cnt 는 $d(x,y)$ 가 Th 이상인 화소의 개수를 의미한다. 또한 Th 는 실험에 사용한 모든 영상에서 고주파 지수가 일정하도록 하는 값을 찾아 8로 설정하였다.

$$\begin{aligned}
 d(x,y) &= |2 * p(x,y) - p(x+1,y) - p(x-1,y)| \\
 Dif &= \{d(x,y) | d(x,y) > Th\} \\
 cnt &= n(Dif) \\
 \text{고주파 지수} &= \frac{1}{cnt} \sum_{k=0}^{cnt} Dif_k
 \end{aligned}$$

(11)

식 (11)에서 구해진 고주파 지수를 통해, 1,000개의 영상을 캡처하여 워터마크 디코딩을 수행한 결과를 그림 12에 그래프로 나타내었다. 대부분의 캡처 영상은 고주파 지수가 17.5~19.5 정도의 값을 가지며, 19.5 이상으로 고주파 성분이 많은 영상과 17.0 이하로 저주파 성분이 많은 영상은 그 숫자가 미미하다. 또한 주파수 지수가 17.0 이하에 해당하는 영상은 비트에러율이 점진적으로 증가한다. 따라서 본 실험에서는 주파수 지수가 17.0 이하일 경우, 워터마크 추출 시 해당 영상을 배제하고 다시 캡처받도록 한다.

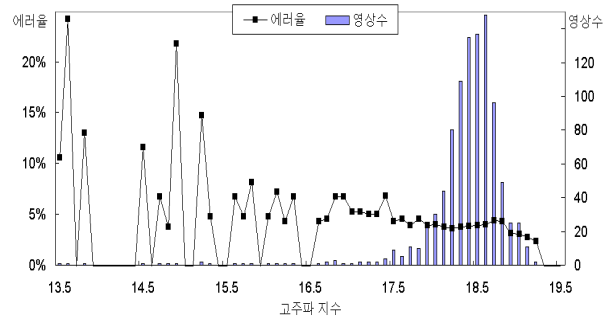


그림 12. 주파수 성분별 영상수와 에러율

4.3 반복적 적응 임계값의 산출

디지털 카메라가 인간 시각 시스템과 최대한 유사하게 하도록 정성적인 출력을 제공한다. 이 때문에 디지털 영상의 프레임 화소값과 캡처된 영상의 프레임 화소값의 차이가 커질 수 있다[5]. 이로 인해 캡처된 영상 중에서는 프레임 화소값이 초기 설정된 프레임 임계값을 벗어나 프레임 영상이 오검출될 수 있다.

이러한 문제를 해결하기 위해 그림 13과 같은 방법으로 프레임 화소 평균을 반복적으로 산출하여 최적의 프레임 임계값을 적용한다. 먼저, 초기 프레임

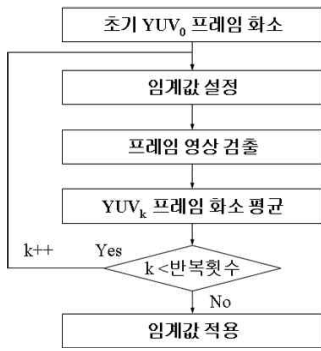


그림 13. 반복적 적응 임계값의 산정 과정

영상에서 프레임으로 인식된 화소의 평균을 계산한다. 이 과정도 캡처 영상을 상하로 16 등분하여 진행된다. 계산된 프레임 화소 평균을 식 (6)을 이용하여 임계값을 설정한다. 다음으로 식 (7)을 이용하여 그림 9와 같이 프레임 영상을 검출하여, 다시 프레임 화소 평균을 계산함으로써 그림 14와 같이 프레임 화소 평균을 찾고, 프레임 임계값을 설정할 수 있다. 반복 횟수는 검출 성능과 수행속도를 고려해 실험적으로 결정한다.

5. 실험결과

본 논문은 워터마크가 삽입된 디지털 영상을 디스플레이하기 위해 삼성 24인치 모니터(2443BWX)를 사용하였으며, 디스플레이된 영상을 캡처하고 영상 처리 및 워터마크 디코딩을 수행하는 스마트폰으로

표 1. 삽입용량 및 화질 비교

실험영상	블록크기	삽입용량	삽입 임계값	PSNR[dB]
Lena	13×6	206	255	42.32
Peppers	13×6	206	270	42.08
Baboon	13×6	206	310	40.53
House	13×6	206	280	40.89
F-16	13×6	206	300	40.52

모토로라사의 아트릭스 M860을 사용하였다. 스마트폰 상에서의 캡처 해상도는 수행속도와 검출성능을 고려하여 640×600으로 설정하였으며, 캡처된 영상에서 중앙부분의 480×480 크기만을 이용하여 프레임 검출 및 디코딩 과정을 수행하였다.

256×256 크기의 표준영상을 웨이브릿 변환 후 블록의 크기는 13x6으로 설정하여 총 206비트의 정보를 삽입하였으며, 실험결과의 비교를 위해 에러정정 코드는 적용하지 않았다. 삽입 임계값은 표 1과 같이 설정하여 각 표준 영상마다 10개의 서로 다른 워터마크로 정보를 삽입하였다. 그 결과 모든 영상에서 PSNR이 40 dB 이상을 나타내어 워터마크 영상의 시각적인 화질 하락을 발견할 수 없었다.

그림 15는 원본 영상과 워터마크 삽입 후의 영상이다. 워터마크가 삽입된 후 최종적으로 영상의 외부에 특정색의 프레임을 추가한다. 이것은 워터마크 추출과정에서 프레임을 검출하여 영상의 왜곡을 보정하기 위한 것이다. 프레임의 색상은 워터마크 영상과

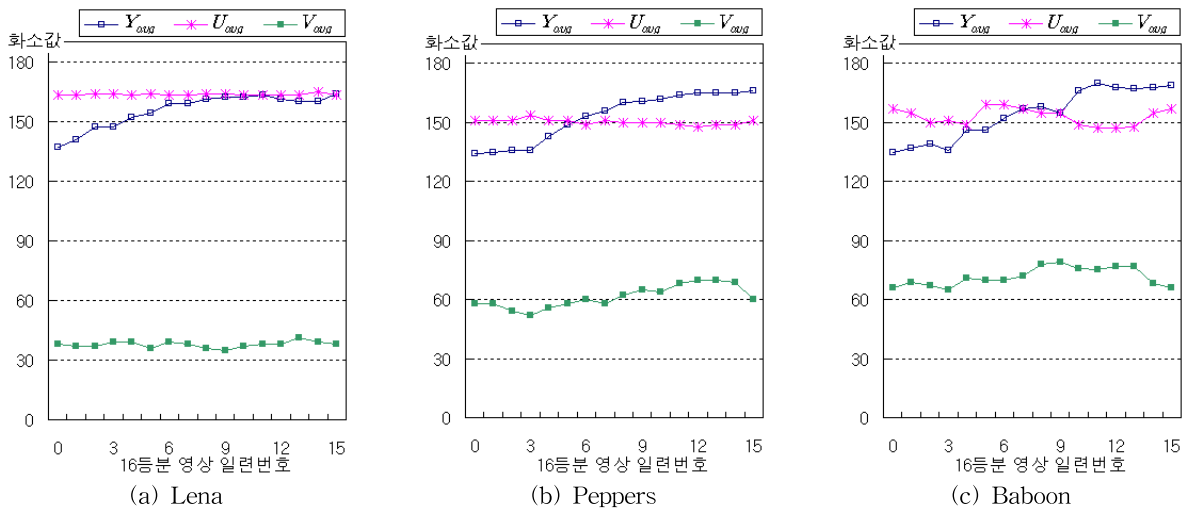


그림 14. 채널별 최종 프레임 화소 평균의 예



(a) 워터마크 삽입 전 (b) 워터마크 삽입 후
그림 15. 워터마크 삽입 영상의 예

영상 외부 배경으로부터 프레임을 검출할 수 있는 색상이라면 가능하다. 본 실험에서는 프레임 검출이 용이하면서도 시각적으로 튀지 않는 색상을 선택하여 YUV 값으로 189, 136, 78을 설정하였다.

스마트폰에서 일반적인 작업은 실시간 처리를 기본으로 한다. 디스플레이-캡처 과정과 워터마크의 추출 또한 스마트폰 사용자는 실시간으로 응답 받기를 요구할 것이다. 따라서 표 2와 같이 워터마크 추출을 위한 각 단계별 수행시간을 나타내었다.

스마트폰은 초당 최대 30장의 사진을 캡처할 수 있으며, 캡처 후 아무런 과정을 행하지 않아도 약 30 ms의 딜레이가 발생한다. 또한 가장 수행시간을 많이 차지하는 과정은 교차점 검출과 왜곡 보정으로 약 40 ms, 32 ms 가 소요되어 전체 과정의 절반 정도를 차지한다. 그리고 임계값 산출은 3회 반복 수행할 때, 약 12 ms 정도 소요된다. 고정 임계값을 사용하는 방법에 비해 전체 수행시간이 10% 정도 추가되어, 실시간 처리에 지장 없이 수행 가능하다.

그림 16은 적응 임계값의 반복횟수에 대한 실험이다. 실험의 방법은 각 표준 영상마다 서로 다른 10개의 워터마크를 삽입한다. 그리고 워터마크가 삽입된

표 2. 워터마크 추출 단계별 수행시간

추출 과정	수행속도	백분율
영상 캡처 및 기타	33.7 ms	23.0 %
블러링 체크	1.7 ms	1.2 %
임계값 산출 (3회 반복)	12.0 ms	8.2 %
프레임 영상 생성	4.7 ms	3.2 %
외곽 제거	9.9 ms	6.8 %
직선 및 교차점 검출	40.1 ms	27.4 %
왜곡 보정	32.3 ms	22.0 %
디코딩 수행	12.1 ms	8.3 %
전체 수행 시간	146.5 ms	100.0 %

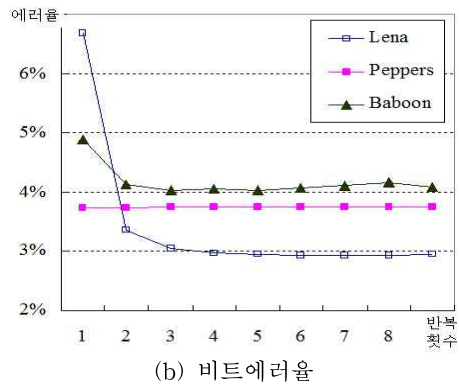
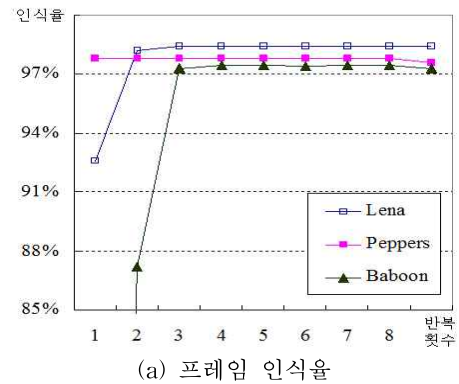


그림 16. 적응 임계값 반복횟수별 비교

10개의 영상을 LCD 모니터에 디스플레이하고, 스마트폰으로 각 100개씩 캡처한다. 이를 통해서 각 표준 영상마다 총 1,000개씩의 캡처 영상을 실험한다. 여기서 프레임 인식율이란 전체 캡처영상 중에 식 (10)에서 D가 4% 이내의 값으로 교차점이 정확히 검출되고, 식 (11)에서 블러링이 발생하지 않은 영상의 비율을 의미한다.

Peppers 영상의 경우 한 번의 임계값 산출만으로도 97%이상의 프레임 인식과 5% 미만의 에러율을 나타낸다. 이것은 초기 설정된 프레임 화소로 산출된 프레임 임계값이 캡처된 Peppers 영상의 실제 프레임 화소와 유사하기 때문이다. Lena와 Baboon 영상의 경우, 반복 횟수가 증가할수록 프레임 인식율은 높아지고, 비트에러율이 감소한다. 이것은 반복적 임계값 산출로 캡처된 영상의 실제 프레임 화소에 근접하게 함을 의미한다. 또한 3번의 반복 이후에는 비트에러율 및 프레임 인식율의 개선이 거의 없기 때문에 본 실험에서는 표 2에서의 검출 속도와 그림 16의 검출 성능을 고려하여 반복적 임계값 산출을 3회 반복 수행한다.

표 3에서 프레임 검출을 위해 고정 임계값을 사용

표 3. 고정 임계값과 적응 임계값의 프레임 인식을 비교(%)

	Lena	Peppers	Baboon
Lena 고정 임계값	93.8	92.1	0.0
Peppers 고정 임계값	53.5	98.0	25.0
Baboon 고정 임계값	1.8	98.0	93.4
반복적 적응 임계값	97.4	98.0	97.3

표 4. 기존 방법과 제안 방법의 비트에러율 비교(%)

	Nakamura 방법[6]	Pramila 방법[7]	제안 방법1	제안 방법2
Lena	20.94	9.16	8.04	2.99
Peppers	17.25	13.80	11.78	3.77
Baboon	33.66	13.87	9.29	4.04
House	16.94	12.81	6.96	4.17
F-16	16.86	16.99	13.25	3.66
평균	21.13	13.33	9.86	3.73

하는 방법과 본 논문에서 제안하는 반복적 적응 임계값을 사용하는 방법을 비교 실험하였다. 각 영상에 맞는 고정 임계값을 실험적으로 찾아서 해당 영상을 검출할 경우, Lena 영상에 맞는 고정임계값을 적용하여 검출하면, Lena와 Peppers 영상에서는 90% 이상에서 프레임을 인식하지만 Baboon의 경우 모든 영상에서 프레임을 인식하지 못하게 된다. 또한 Peppers 영상에 맞는 고정임계값을 적용하면, Peppers 영상만 프레임 인식율이 98%를 나타내었다. Baboon 영상에 맞는 고정임계값을 적용하면, Peppers와 Baboon 영상에서만 프레임 인식율이 90% 이상을 나타내었다. 반면, 제안한 방법의 경우에는 반복적 임계값 산출을 통해 캡처된 해당 영상에 맞게 프레임 화소를 반복적으로 계산하기 때문에 캡처된 영상 중에서 97% 이상에서 프레임을 정확히 인식하였다.

마지막으로 표 4에서 제안한 방법의 워터마크 추출 성능 비교를 위해 사인 패턴을 삽입하는 Nakamura 방법[6]과 웨이브릿 변환영역에서 대역확산 방식으로 워터마킹을 하는 Pramila의 방법[7]을 비교 실험하였다. 다만 기존 방법들이 프레임 검출에 대한 자세한 설명이 없음으로 인해 프레임 검출 방법은 제안한 방법을 따랐으며, 실질적인 워터마킹 추출 성능만을 비교하였다. 또한 기존 방법의 워터마크 삽입의 강도와 삽입 용량은 표 1에 나타난 제안 방법과 같게 설정하였다.

Nakamura 방법은 기존 논문에서 적은 삽입 용량에 많은 반복 삽입으로 성능을 유지한다. 따라서 제안 방법과 같은 용량과 화질로 워터마킹 할 경우, 비트에러율이 평균 20%가 넘게 나타났다. Pramila의 방법 또한 대체적으로 10% 이상의 비트에러율이 발생하였다.

제안한 방법은 워터마크 삽입 과정에서 식 (4)를 만족할 수 있도록 삽입 임계값을 설정하여 반복적으로 삽입하는 제안 방법1과 제안 방법2에 추가하여 웨이브릿 변환의 대역별 특성을 이용하여 식 (3)과 같이 LH 대역에서 랜덤 시퀀스 $S(n,m)$ 를 수직 방향으로 쌍을 이루어 1, -1이 배치되게 구성하여 워터마킹 하는 제안 방법2를 같이 실험하였다.

먼저 제안 방법1의 경우, 기존 방법에 비해 평균 약 26% 이상 개선되어 비트에러율이 10% 이하를 나타내었다. 또한 제안 방법2의 경우, 모든 표준 영상에서 비트에러율이 5% 이하를 나타내어 기존 방법보다 평균 약 70% 이상 향상되었다.

6. 결 론

본 연구는 디지털 콘텐츠의 모바일 활용에 필요한 보안문제를 디지털 워터마킹의 관점에서 해결하고자 하는 디스플레이-캡처 기반의 모바일 워터마킹에 관한 연구이다. 먼저 디지털 영상을 LCD 모니터에 디스플레이 하고, 스마트폰의 카메라로 캡처하는 과정에서 특성을 조사하여 분석하고, 프레임 검출 성능을 향상시킬 수 있는 방법을 제시하였다. 또한 웨이브릿 변환 영역에서 주파수 대역의 특성을 이용하여 랜덤 시퀀스가 쌍을 이루게 구성하였다. 그리고 각 블록의 CCS를 계산하고 삽입 임계값을 이용하여 반복적으로 워터마크를 삽입함으로써 워터마크 추출 성능을 향상시킬 수 있었다.

실험결과, 제안한 방법은 256×256 크기의 표준 영상에 206 bits의 정보를 삽입하여 PSNR이 40 dB 이상을 유지했으며, 기존 방법인 Pramila 방법에 비해 비트에러율이 평균 70% 이상 개선됨을 알 수 있었다. 향후 연구로는 실제 응용을 위해 프레임 없이 왜곡을 보정하여 워터마크를 추출할 수 있는 방법에 대한 연구가 필요하다.

참 고 문 헌

[1] I.J. Cox, M.L. Miller, and J.A. Bloom, *Digital*

Watermarking, Morgan Kaufmann Publishers, San Francisco, Calif., 2002.

[2] 김범수, 최재각, “불변의 무게중심을 이용한 영상 정규화에 기반한 기하학적 공격에 강인한 워터마킹,” 한국정보과학회논문지, 제31권, 제3호, pp. 243-251, 2004.

[3] D. He and Q. Sun, “A Practical Print-scan Resilient Watermarking Scheme,” *ICIP2005*, Vol. 1, pp. 257-260, 2005.

[4] 최윤희, 최태선, “블록 DCT와 영상 정규화를 이용한 회전, 크기, 이동 변환에 견디는 강인한 로고 삽입방법,” 한국정보보호학회논문지, 제15권, 제5호, pp. 93-103, 2005.

[5] John Stach, Trent Brundage, Brett Hannigan, Brett Bradley, Tony Kirk, Hugh Brunk, and Digimarc Corp. “On the Use of Web Cameras for Watermark Detection,” *Security and Watermarking of Multimedia Contents IV*, Vol. 4675, pp. 611-620, 2002.

[6] Takao Nakamura, Atsushi Katayama, Masashi Yamamuro, and Noboru Sonehara, “Fast Watermark Detection Scheme for Camera-equipped Cellular Phone,” *Proc. of the International Conference on Mobile and Ubiquitous Multimedia*, Vol. 83, pp. 101-108, 2004.

[7] Anu Pramila, Anja Keskinarkaus, and Tapio Seppänen, “Watermark Robustness in the Print-Cam Process,” *Proc. of the IASTED International Conference on Signal Processing Pattern Recognition and Applications*, pp. 60-65, 2008.

[8] 최성, 박석천, “모바일 콘텐츠 서비스 현황과 발전 방향,” 한국정보처리학회지, 제9권, 제2호, pp. 43-50, 2002.

[9] Bo-Ho Cho, Jong-Wook Bae, Jae-Young Lee, Jeng-Hae Youn, and Sung-Hwan Jung, “A Study on Image Distortion for Display-capture based Mobile Watermarking,” *2006 International Conference on Asian Language Processing and Information Technology (ALPIT2006)*, Vol. 5, pp. 249-253, 2006.

[10] Jong-Wook Bae and Sung-Hwan Jung, “A New Mobile Watermarking Scheme Based on Display-capture,” *Journal of Korea Multimedia Society*, Vol. 12, No. 6, pp. 815-823, 2009.

[11] R. Sukthankar, R.G. Stockton, and M.D. Mullin, “Self-Calibrating Camera-Assisted Presentation Interface,” *Proc. of International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2000.

[12] 배종욱, 정성환, “디스플레이-캡처 영상을 위한 웨이브릿 기반의 모바일 워터마킹,” 한국정보과학회 학술심포지엄 유비쿼터스 컴퓨팅과 웹 정보기술, 제5권, 제3호, pp. 115-119, 2011.

배 종 욱



2001년 창원대학교 컴퓨터공학과(공학사)
 2005년 창원대학교 대학원 컴퓨터공학과(공학석사)
 2005년~현재 창원대학교 대학원 컴퓨터공학과 박사과정

관심분야 : 워터마킹, 멀티미디어 정보보호, 영상처리 등

정 성 환



1979년 경북대학교 전자공학과(공학사)
 1983년 경북대학교 대학원(공학석사-정보통신)
 1988년 경북대학교 대학원(공학박사-영상처리)

1983년~1985년 한국전자통신연구소 연구원
 1986년 전자계산기기술사, 1992년 정보처리기술사
 2003년 정보시스템감리사
 1993~1994년 Univ. of California(UCSB) Post-Doc.
 1999~2000년 Colorado School of Mines 연구교수
 2008~2009년 Univ. of Missouri(UMKC) 방문교수
 1988년~현재 창원대학교 컴퓨터공학과 교수
 관심분야 : 영상처리, 컴퓨터비전 및 패턴인식, 멀티미디어 정보보호