Checkmark를 이용한 디지털 워터마킹 평가 연구

강현호, 임쌍학, 박지환
부경대학교 전자계산학과
부경대학교 교육대학원 전산교육전공,
부경대학교 전자컴퓨터정보통신공학부

A Study on Digital Watermarking Evaluation Using Checkmark

Hyun-Ho Kang, Ssang-Hak Ihm, Ji-Hwan Park
Dept. of Computer Science, Pukyong National University
Dept. of Computer Education, Pukyong National University
Div. of Electronic, Computer & Telecom. Eng., Pukyong National University

요 약

디지털 워터마킹 기법 평가를 위한 다양한 시도가 행해지고 있다. 특히 디지털 영상에 대한 공격기법을 도구화한 것중에 Matlab으로 구현된 Checkmark가 개발되어 있다. 특히 디지털 영상에 대한 워터마킹 수행모듈(삽입, 추출포함)은 평가를 위한 공격모듈과 별개로 분리할 수밖에 없는 설정이었다. Checkmark는 특별히 워터마킹의 추출과정을 사용자가 설정하여 Checkmark 수행 모듈에 포함시킬 수 있도록 배려하였다. 이러한 시도는 새로운 워터마킹 알고리즘의 개발 과정에 상당히 효율적인 부분이라고 할 수 있다. 본 논문은 이러한 점을 중심으로 Checkmark의 사용자 설정 모듈을 소개하고자 한다.

1. 서 론

디지털화된 다양한 컨텐츠의 저작권 보호를 위한 방법으로 디지털 워터마킹(digital watermarking)이 연구되고 있다. 디지털 워터마킹은 컨텐츠 제공자의 정보를 나타내는 워터마크(watermark)를 보호하고자 하는 컨텐츠에 삽입시키고 추후 저작권의 침해가 있을 경우 컨텐츠 제공자의 것임을 증명할 수 있다.

디지털 컨텐츠의 종류나 응용에 따라 디지털 워터마킹의 요구사항도 달라져야 하지만 기본적인 요구사항은 다음과 같다[1].

- ▷ 무지각성(imperceptibility) : 워터마킹 전후의 컨텐츠를 지각적으로 구별할 수 없어야 한다.
- ▷ 강인성(robustness) : 워터마킹된 컨텐츠가 변경이 되더라도 워터마크를 얻을 수 있어야 한다.
- ▷ 용량(capacity) : 컨텐츠에 삽입될 워터마크는 응용에 따라 일정한 양을 갖고 있어야 한다. 복사 방지와 같은 응용에서는 1비트만으로도 충분할 수가 있다.

본 연구는 SEDICA 지원에 의해 수행되었음

위의 기본적인 요구사항과 다른 요구사항에 대한 평가를 위한 다양한 시도의 예로 알려진 몇개의 평가툴이 있다[2][3]. 그 중에서 Matlab으로 구현된 Checkmark의 내용을 통해 디지털 워터마킹 평가를 검토하기로 한다. 특히, Checkmark는 디지털 컨텐츠의 종류 및 응용에 따라 그룹화 하여 보다 효율적인 평가의 척도를 위한 노력을 하고 있다[4-7].

본 논문은 2장에서 Checkmark의 구성을 소개하고, 3장에서 사용자가 워터마킹 추출모듈을 Checkmark에 설정하는 방법에 대해서 검토한다. 4장에서 실제영상에 대해 간단한 실험을 한 후 5장에서 결론을 맺는다.

2. Checkmark 소개

Checkmark는 워터마킹 기술의 평가를 위한 벤치마킹 툴로서 2001년 12월에 버전 1.2가 개발되어 있다. 다양한 워터마킹 기술에 대한 수요의 증가함에 따라 이러한 벤치마킹 툴의 요구 또한 증가하고 있는 것이 사실이고 이러한 툴간의 평가 또한 필요할 것으로 보인다.

2.1 구성

Checkmark에서는 워터마킹 공격항목을 4개의 레벨로 구분하여 분류하고 있다. 첫 번째 레벨은 공격그룹(attack group)으로 그림1 과 같이 4개의 항목으로 구성되어 있다. 두 번째 레벨은 공격형태(attack type)로 그림1에서 Removal 그룹은 Denoising, Compression 등의 여러 공격형태로 구성되어 있다. 세 번째 레벨은 공격클래스(attack class)로 Denoising 타입은 ML (maximum likelihood), MAP(maximum posteriori probability)와 같은 클래스로 구성되어 있다. 네 번째 레벨은 개별공격(individual attack)으로 각 클래스는 하나 혹은 그 이상의 개별 공격으로 구성되어 있다. 예를 들면 MAP 공격 클래스는 Wiener filtering, soft thresholding 등의 개별공격으로 구성되어 있다.

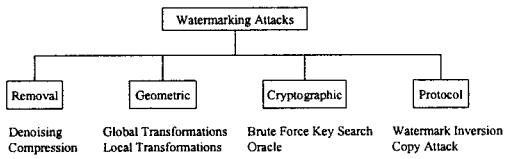
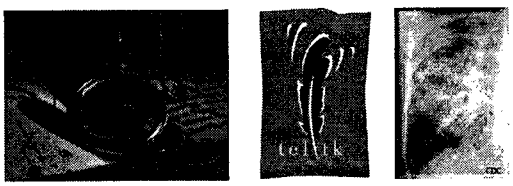


그림1. 워터마킹 공격 분류

또한, Checkmark는 응용에 따라서 공격방법을 6가지로 분류하고 있다. 다시말해 22개의 공격 클래스를 적용 워터마킹 영상의 응용에 따라 6개의 응용항목에 적절히 분류하고 있다.

(1) nongeometiric (46개의 개별공격)
(2) copyright (382개의 개별공격)
(3) banknote (178개의 개별공격)
(4) logo (77개의 개별공격)
(5) medical (45개의 개별공격)
(6) video (193개의 개별공격)

위 응용의 분류에서 medical 영상과 logo 영상의 경우를 제외한 나머지는 같은 분류의 영상으로 처리하고 있다.



일반영상 로고영상 의료영상

그림2. 응용에 따른 워터마킹 영상

2.2 실행방법

Checkmark는 Matlab으로 구현된 프로그램으로 크게 4개의 중요한 파일함수로 구성되어 있다.

(1) getconfig.m (경로설정)
워터마크된 영상 혹은 공격된 영상의 경로설정, 응용종류 설정, 워터마킹 기법 설정 등을 수행한다.

(2) runCheckmark.m (공격수행)
워터마크된 각종 영상에 대해서 응용종류에 따라 공격을 수행하여 JPEG파일 포맷으로 출력한다.

(3) testDetector.m (XML파일 포맷으로 결과출력)
이 파일함수에서 워터마킹의 추출모듈을 사용자가 삽입하여 수많은 공격영상에 대한 결과처리를 일괄적으로 수행할 수 있다. 3장에서 이 영역에 대해서 자세히 다루기로 한다.

(4) parseresults.m (HTML문서로 결과를 출력)
testDetector.m에서 출력된 XML파일의 결과를 바탕으로 HTML문서로 출력한다.

3. 워터마킹 추출 모듈 구현 방법

Checkmark에서는 워터마킹 기법의 평가를 위해 [8]의 소스를 참조하고 있다. 본 논문에서는 이 소스에서 소개되고 있는 Cox의 기법을 예로 들어 실제로 Checkmark 수행과정에 이 모듈을 삽입하는 방법을 소개하기로 한다.

표1. Cox 워터마킹 수행 스크립트

gen_cox_sig -n 1000 -o cox.sig
wm_cox_e -s cox.sig -o cox_lena.pgm lena.pgm
wm_cox_d -s cox.sig -i lena.pgm -o cox.wm cox_lena.pgm
cmp_cox_sig -s cox.sig cox.wm

위의 블록내용은 순서대로 워터마크를 생성, 삽입, 검출, 비교를 수행하는 스크립트이다. 주의 할 것은 Checkmark에서 도입한 이 소스의 입출력은 PGM파일 포맷의 512*512 크기 8비트 그레이 스케일만을 한정하고 있다.

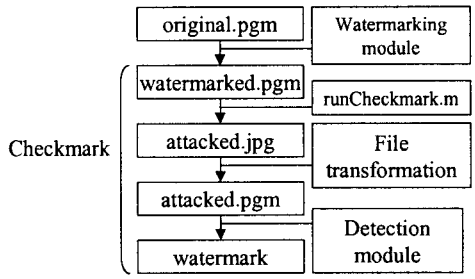


그림3. Checkmark 적용범위

Checkmark에서는 testDetector 함수에서 호출하는 executedetector 함수를 사용자에게 설정하도록 하여 워터마크의 유무를 1과 0으로 반환하도록 한다. 표1의 추출 스크립트를 이 함수에 설정하여 그림3에서의 Detection module을 Checkmark 영역으로 포함시킬 수 있다. 이에 앞서 runCheckmark 함수를 통해 수행된 공격된 영상의 JPEG포맷을 PGM포맷으로 변경시켜야 한다. 본 논문에서는 PNM Toolbox[9]를 사용하여 그림3의 File transformation 모듈을 수행하고 있다. 표2에서 executedetector 함수에서 실제 수행할 수 있는 File transformation 모듈 코드와 Watermark detection 모듈 코드를 보이고 있다.

표2. File Transformation & Watermark detection

```
YY=imread(fname);
pnmwrite(YY,'007.pgm');
com1=sprintf('!wm_cox_d -n 1000 -s cox.sig -i im%d.pgm -o cox.wm 007.pgm',i)
eval(com1)
com1_1=sprintf('!cmp_cox_sig -o corr_val -s cox.sig cox.wm')
eval(com1_1)
fid123=fopen('corr_val','r');
abc=fgetl(fid123);
abc=str2num(abc);
fclose(fid123);
if abc >0.2
   detectionresult=1;
else
   detectionresult=0;
end
```

4. 실험 및 결과

실험에 사용된 영상은 그림4의 5가지로 8비트 그레이 스케일 영상이다.

그림4. 실험에 사용된 영상(8-bit grayscale)

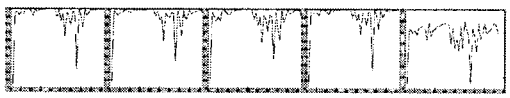


그림5. 각 영상에 대한 추출 상관도

워터마킹 방법은 Cox방법이고 Checkmark 응용항목은 nongeometric 항목으로 하나의 영상에 46개 공격이므로 총 230개의 공격이 수행되었다. 추출과정은 상관도 값이 0.2보다 큰 것을 1로 하여 총 230개의 공격에 대한 추출 성공 비율을 보인다. 그림5에서 가로축은 개별공격의 개수이고 세로축은 상관도 값을 보이고 있는데 거의 대부분이 0.2이상을 보이고 있다. 다른 워터마킹 방법에 대한 공격 평가는 [7]에서 자세히 살펴볼 수 있다. 표3은 230개의 개별공격에 대해 229개의 워터마크(상관도 0.2이상)가 추출되었음을 보이고 있다. 실제 Checkmark에서 제공하고 있는 데이터는 Cox방법에 대해서 90%를 보여주고 있는데 이러한 결과의 차이는 테스트되고 있는 영상의 종류에 따라 차이가 있을 것이다.

표3. 추출 성공 비율(nongeometric)

Attacks/Images	im1	im2	im3	im4	im5	Avg
reSample(1)	1 100%	1 100%	1 100%	1 100%	1 100%	100%
Wavelet(10)	10 100%	10 100%	10 100%	10 100%	10 100%	100%
Remodulation(4)	4 100%	4 100%	4 100%	4 100%	4 100%	100%
ML(7)	7 100%	7 100%	7 100%	7 100%	7 100%	100%
MAP(6)	6 100%	6 100%	6 100%	6 100%	6 100%	100%
JPEG(12)	12 100%	12 100%	12 100%	12 100%	12 100%	100%
Filtering(3)	3 100%	3 100%	3 100%	3 100%	3 100%	100%
Copy(1)	1 100%	1 100%	1 100%	1 100%	1 100%	100%
ColorReduce(2)	2 100%	2 100%	2 100%	2 100%	1 50%	90%
Average(230)	100%	100%	100%	100%	94%	99%

표3에서 보이는 각 공격 클래스에 사용된 개별공격을 부록에 보인다.

5. 결 론

워터마킹 기법에 대한 공격 벤치마킹 툴인 Checkmark의 기본 구성을 살펴보고, Checkmark에서 도입하고 있는 Meerwald의 소스를 적용하여 직접 이 코드의 추출모듈을 Checkmark의 소스에서 구현하였다. 최근에는 Stirmark의 최근 버전에도 각 응용에 따른 프로파일을 만들어 효율적인 접근을 시도하고 있으나 워터마크의 추출과정의 수행은 여전히 복잡한 작업이 요구되고 있다. 본 논문에서 제안하는 Checkmark의 추출모듈 삽입 방법을 활용한다면 다양한 공격모델의 평가에 있어서 더욱 효율적인 작업의 수행이 가능할 것이다.

[참고문헌]

[1] G. C. Langelaar, I. Setyawan and R. L. Lagendijk, "Watermarking Digital Image and Video Data: A State-of-the-art Overview", IEEE Signal Processing Magazine, Vol.17, No.5, pp.20-46, September 2000.

[2] http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/

[3] http://poseidon.csd.auth.gr/optimark/

[4] http://watermarking.unige.ch/Checkmark/

[5] S. Pereira, S. Voloshynovskiy, M. Madueno, S.Marchand-Maillet and T.Pun, "Second generation Benchmarking and Application Oriented Evaluation", Information Hiding Workshop III, April 2001.

[6] S. Voloshynovskiy, S. Pereira, V. Iquise and T. Pun, "Attack modelling : Towards a Second Generation Watermarking Benchmark", Signal Processing, Special Issue : Information Theoretic Issues in Digital Watermarking, May 2001.

[7] P.Meerwald and S.Pereira, "Attacks, Applications and Evaluation of known Watermarking Algorithms with Checkmark", SPIE, Vol.4675, Jan. 2002.

[8] http://www.cosy.sbg.ac.at/~pmeerw/Watermarking

[9] http://home.online.no/~pjacklam/matlab/software/pnm/index.html

[부록] - nongeometric 응용에 사용된 개별공격항목

▷reSample(1)

subGroup	CType	CFactor	down sample	up sample
sampledownup1	J	100.00	0.75	1.33
sampledownup2	J	100.00	0.50	2.00
sampledownup3	J	100.00	0.75	1.30
sampledownup4	J	100.00	0.50	1.90

▷Wavelet(10)

subGroup	CType	CFactor
waveletcompression1	W	0.10
waveletcompression1	W	0.20
waveletcompression1	W	0.30
waveletcompression1	W	0.40
waveletcompression1	W	0.50
waveletcompression1	W	0.60
waveletcompression1	W	0.80
waveletcompression1	W	1.50
waveletcompression1	W	3.50
waveletcompression1	W	8.00

▷Remodulation(4)

subGroup	CType	CFactor	window
dpr1	J	100.00	3.00
dpr2	J	100.00	5.00
dprcorr1	J	100.00	3.00
dprcorr2	J	100.00	5.00

▷ML(7)

subGroup	CType	CFactor	window
medfilt1	J	100.00	2.00
medfilt2	J	100.00	3.00
medfilt3	J	100.00	4.00
trimmedmean1	J	100.00	3.00
trimmedmean2	J	100.00	5.00
medpoint1	J	100.00	3.00
midpoint2	J	100.00	5.00

▷MAP(6)

subGroup	CType	CFactor	window
hardthresh1	J	100.00	3.00
hardthresh2	J	100.00	5.00
softthresh1	J	100.00	3.00
softthresh2	J	100.00	5.00
wiener1	J	100.00	3.00
wiener2	J	100.00	5.00

▷JPEG(12)

subGroup	CType	CFactor
jpegcompression1	J	10.00
jpegcompression1	J	15.00
jpegcompression1	J	25.00
jpegcompression1	J	30.00
jpegcompression1	J	40.00
jpegcompression1	J	50.00
jpegcompression1	J	60.00
jpegcompression1	J	75.00
jpegcompression1	J	80.00
jpegcompression1	J	85.00
jpegcompression1	J	90.00
jpegcompression1	J	100.00

▷Filtering(3)

subGroup	CType	CFactor	window
gaussian1	J	100.00	3.00
gaussian2	J	100.00	5.00
sharpening1	J	100.00	5.00

▷Copy(1)

subGroup	CType	CFactor
copy1	J	100.00

▷ColorReduce(2)

subGroup	CType	CFactor
dither1	J	100.00
thresh1	J	100.00