

단어 분류에 기반한 텍스트 영상 워터마킹 알고리즘

(An Algorithm for Text Image Watermarking based on Word Classification)

김 영 원 [†] 오 일 석 ^{**}
(Young-Won Kim) (Il-Seok Oh)

요 약 본 논문은 단어 분류에 기반한 새로운 텍스트 영상 워터마킹 알고리즘을 제안한다. 간단한 특징을 이용하여 단어를 K 개로 분류한다. 이웃한 몇 개의 단어들을 조합하여 세그먼트를 구성하고, 세그먼트에 속한 단어들의 부류에 의해 세그먼트 또한 분류된다. 각 세그먼트에 동일한 양의 신호가 삽입된다. 신호 삽입은 세그먼트 부류가 갖는 단어 간 공백의 통계값을 조작함으로써 이루어진다. 몇 가지 기준에 따라 기존 단어 이동 알고리즘과의 주관적인 비교가 제시된다.

키워드 : 텍스트 워터마킹, 단어와 세그먼트 분류, 단어 이동 알고리즘, 단어간 공백의 통계적 분포

Abstract This paper proposes a novel text image watermarking algorithm based on word classification. The words are classified into K classes using simple features. Several adjacent words are grouped into a segment, and the segments are also classified using the word class information. The same amount of information is inserted into each of the segment classes. The signal is encoded by modifying some inter-word spaces statistics of segment classes. Subjective comparisons with conventional word-shift algorithms are presented under several criteria.

Key words : text watermarking, word and segment classification, word-shift algorithm, inter-word space statistics

1. 서 론

인터넷의 발달로 인한 콘텐츠의 증가와 함께 저작권 보호의 필요성이 대두되고 있다. 이를 해결하기 위한 방법으로는 암호화, 인증 시스템, 그리고 워터마킹 등의 기술이 있으며, 이를 위한 연구의 비중 또한 갈수록 커지고 있다. 워터마킹이란 영상, 비디오, 소리, 텍스트 등의 디지털 저작물에 소유권을 나타내는 신호를 삽입하는 기술이다[1]. 현재 워터마킹에 관한 많은 연구가 활발히 이루어지고 있으며 응용 분야와 응용 사례 또한 증가하고 있다. 예로 e-book[2], 전자도서관[3], 전자정부[4] 등의 사업을 들 수 있으며, 이러한 사업에서는 중요 문서를 영상 형태로 보관하고 서비스하는 사례가 증가하고 있다. 이러한 응용 사업과 더불어 문서 영상에

대한 저작권 보호 문제와 인증 시스템의 중요성이 부각되고 있다. 그러나 기존의 워터마킹 연구는 정지 영상과 비디오에 치우쳐 있어 텍스트를 위한 워터마킹 알고리즘의 개발과 연구는 미비한 실정이다.

문서 영상은 전경과 배경이 분리되어 정지영상을 위한 일반적인 워터마킹 알고리즘을 사용할 경우 심한 일그러짐이 나타난다[5]. 따라서 문서 영상의 특징을 제대로 파악하고, 이에 따라 워터마킹 알고리즘을 설계해야 한다. 이미 발표된 텍스트를 위한 워터마킹 알고리즘은 이미지 형태의 문서를 위한 방법과 ASCII 파일로 저장된 문서를 위한 워터마킹 알고리즘으로 분류할 수 있다.

이미지 형태의 문서를 위한 알고리즘으로는 줄/단어 이동 알고리즘과 특정 코딩 알고리즘 그리고 히스토그램 영역 알고리즘과 명도 조절 알고리즘이 있다. 줄 이동 알고리즘은 인코딩 규칙에 의한 특정 줄을 위로 이동하거나 아래로 이동하여 각 줄 사이의 간격을 신호로 삽입하며, 단어 이동 알고리즘은 같은 방법으로 특정 단어를 왼쪽이나 오른쪽으로 이동하여 신호를 삽입한다

· 이 논문은 한국전자통신연구원(ETRI)의 지원으로 이루어졌습니다.

[†] 학생회원 : 전북대학교 전자정보공학부
ywkim@dahong.chobuk.ac.kr

^{**} 종신회원 : 전북대학교 전자정보공학부
isoh@chonbuk.ac.kr

논문접수 : 2004년 7월 16일

심사완료 : 2005년 6월 21일

[6-8]. 또한 단어 이동 알고리즘으로 Huang은 삼입할 신호를 sine 파의 위상, 진폭, 주파수에 인코딩 하였으며, 신호 삼입 방법으로 글자 사이의 폭을 조절하여 단어의 간격이 sine 파에 따르도록 하였다[9]. 특징 코딩 알고리즘은 문자를 구성하는 획의 특징을 추출하거나 조작하는 기법으로 Brassil은 문자 b, d, h등의 긴 획을 짧게 또는 길게 조작하는 방법으로 특징 코딩 워터마킹을 제안하였다[10-12]. Amano는 인코딩 규칙에 따라 문자의 특정 획을 선택하여 획의 굵기를 조작하는 알고리즘을 제시하였다[13]. 히스토그램 영역 알고리즘과 명도 조절 알고리즘은 명암 값을 갖는 문서 영상을 위한 알고리즘이다. 김영원은 한 문서 영상 내에 일정 크기의 블록들은 모두 같은 히스토그램 분포를 나타내는 특성을 이용하여 기준이 되는 영역과 조작 영역을 정하고, 기준이 되는 영역과 조작된 영역의 히스토그램이 같으면 '0' 다르면 '1'로 신호를 추출하였다[14]. Bhattacharjya는 두 텍스트 영역의 평균 명도를 조절하여 정보를 숨기는 방법을 제시하였다[15].

ASCII 파일을 위한 워터마킹 알고리즘으로 특정 단어나 구를 비슷한 어휘로 바꾸는 방법[16,17]과 공백(white space)은 인쇄된 종이에 나타나지 않는다는 특성을 이용하여 텍스트 내에 공백의 개수를 신호로 삼입하는 방법[16]이 있다. 비슷한 어휘로 바꾸는 방법은 문장이 어색해지거나 애매한 경우를 발생시키며, 시와 같이 압축적이고 많은 의미를 포함하고 있는 문장의 경우에는 원본을 크게 훼손하는 결과를 초래한다. 공백의 개수를 신호를 삼입하는 방법은 비인지는 우수하나 포맷 변환 또는 워드 프로세서의 작업 등에 의해 신호가 사라지게 되어 강인성에 문제점이 있다. 이렇듯 ASCII 파일은 신호가 눈에 띄지 않게 조작하기 어려우며 또한 신호를 숨길 수 있는 공간이 부족하기 때문에 ASCII 파일을 위한 워터마킹 연구는 영상에 비하여 어렵고 연구도 적다.

이 논문은 단어 분류에 기반한 텍스트 영상 워터마킹 알고리즘을 제안한다. 기존의 방법은 특정 위치의 단어나 줄을 일정한 양만큼 이동하나, 본 논문의 방법은 단어들의 상대적인 길이를 비교하여 단어들을 분류하고 단어 간 공백의 크기를 비교하여 단어를 이동한다. 텍스트 영상내의 단어들의 분포를 살펴보면, 각 단어 부류의 위치는 랜덤하게 발생하고 각 단어 부류의 발생 빈도도 다르다. 이러한 특징을 이용하여 신호의 위치와 양을 전체 영상에 골고루 분포하도록 신호 삼입 단위인 세그먼트를 정의하였다. 세그먼트는 이웃한 단어들의 조합으로 구성되고 신호는 세그먼트내의 단어의 왼쪽과 오른쪽 공백 크기의 통계값을 조작하여 삼입된다. 통계값의 조작은 비인지와 강인성에서 여러 가지 장점을 가지고

있음을 설명한다.

2. 워터마킹 알고리즘

이 장에서는 텍스트 영상에서 단어를 K 개로 분류하는 방법과 분류된 단어들을 조합하여 세그먼트를 구성하는 방법에 대해 설명한다. 또한 각 세그먼트에 포함된 단어 간 공백들의 통계를 이용하여 신호를 삼입하고 추출하는 방법을 설명한다.

2.1 기본 원리

이 알고리즘의 기본 아이디어는 단어를 K 개로 분류하고 각 부류에 p bits의 신호를 삼입하는 것이다. 단어 분류(word classification)는 단어 영상으로부터 적당한 특징을 추출하여 수행한다. 총 N 개의 단어가 있을 때, i 번째 단어 영상을 w_i 라 하고, w_i 가 속한 부류를 $C(w_i)$ 라 표기하자. $S(k)$ 는 부류 k 에 속하는 단어의 집합을 나타내고 $|S(k)|$ 는 $S(k)$ 에 속하는 단어 개수를 나타낸다.

부류 k 에 신호를 삼입하기 위해서는 $S(k)$ 에 속하는 단어들의 오른쪽 공백과 왼쪽 공백의 크기 분포를 이용한다. 가장 간단한 방법은 $S(k)$ 의 왼쪽 공백 크기 평균과 오른쪽 공백 크기 평균을 각각 m_k^l 과 m_k^r 이라 할 때 다음과 같은 규칙으로 1 bit를 삼입하는 것이다.

if ($m_k^r \leq m_k^l$), signal 0

otherwise, signal 1

이때 원하는 신호에 해당하는 통계 분포를 만들기 위해서는 $|S(k)|$, ($1 \leq k \leq K$)가 충분히 크고 부류간 균형을 이루어야 한다. 균형을 이루게 하기 위해서는 분류기가 사용하는 특징을 신중하게 설계해야 한다. 또한 특징은 계산이 간편해야 하며 손상이 된 상태에서도 분류기 오류를 최소화 할 수 있는 조건을 만족해야 한다.

일반적으로 통계 분포를 사용하여 p bits를 삼입할 수 있다. 예를 들어, Gaussian 분포를 만들어 (μ, σ) 를 이용하여 여러 bits를 삼입할 수 있다. k 번째 부류의 왼쪽 공백의 통계 분포와 오른쪽 공백의 통계 분포를 Ω_k^l 과 Ω_k^r 이라 하자. 어떤 부류에 해당하는 신호를 삼입하기 위해서는 Ω_k^l 과 Ω_k^r 을 변경할 필요가 있다. 이를 달성하기 위해서는 $S(k)$ 에 속하는 단어들을 좌우로 이동해야 한다. 그러나 단어의 오른쪽 또는 왼쪽 공백 크기가 변동되면 바로 이웃한 단어의 왼쪽 또는 오른쪽 공백의 영향을 미치게 된다. 이렇듯 단어 부류별 이동은 다른 부류에 영향을 주므로 모든 부류가 자신에게 부여된 신호를 삼입할 수 있도록 광역적으로 단어 이동을 조정하는 알고리즘이 필요하다.

2.2 단어와 세그먼트 분류

이 논문에서 제안하는 방법은 단어 분류를 위해 단순

하고 신뢰성 높은 특징을 사용한다. 이웃한 단어 n 개를 하나의 그룹으로 하여 이를 세그먼트라 하고, 단어 부류가 아니라 세그먼트 부류를 중심으로 신호를 삼입한다. 이렇게 함으로써 단어 이동을 결정하는 알고리즘이 단순해지고 단어 이동시 발생하는 부류 간 영향도 없다.

단어 w_i 의 분류를 위한 특징으로 w_i 의 길이 (이를 $l(w_i)$ 라 함), 왼쪽 단어의 길이 ($l(w_{i-1})$), 그리고 오른쪽 단어의 길이 ($l(w_{i+1})$)를 이용한다. 단어를 추출하는 대상은 영상이므로 길이는 픽셀 수이다. 이를 이용하여 w_i 를 두 개의 부류(표 1) 또는 네 개의 부류(표 2)로 구분할 수 있다. 표 1의 2 부류 경우는, 단어를 중심으로 왼쪽 단어가 오른쪽 단어보다 길면 중심 단어는 클래스 0, 반대로 왼쪽 단어보다 오른쪽 단어가 더 길면 중심 단어는 클래스 1로 분류한다. 그림 1은 단어가 2부류로 분류된 경우를 보여준다.

표 1 단어의 2 부류

조건	w_i 의 부류
$l(w_{i-1}) > l(w_{i+1})$	0
$l(w_{i-1}) < l(w_{i+1})$	1

표 2 단어의 4 부류

조건*	w_i 의 부류
$a \geq b \ \&\& \ c \geq d$	00(0)
$a \geq b \ \&\& \ c < d$	01(1)
$a < b \ \&\& \ c \geq d$	10(2)
$a < b \ \&\& \ c < d$	11(3)

* $a=l(w_{i-2}), b=l(w_{i-1}), c=l(w_{i+1}), d=l(w_{i+2})$

표 1과 표 2에서 알 수 있듯이 단어는 이웃한 단어의 상대적 길이에 의해서 분류된다. 비교 기준을 사용하는 것은 절대적인 크기 값으로 분류하는 것보다 신뢰성이 높으며, 텍스트 영상의 종류나 문자의 폰트와 크기에 일반적으로 적용할 수 있기 때문이다. 또한 단어의 길이는

랜덤하게 발생하기 때문에 각 단어 부류는 랜덤한 위치에 고르게 발생하며 각 단어 부류의 균등한 발생은 신호 삼입을 위해 중요하다. 각 줄에 첫 번째 단어(w_1)와 마지막 단어(w_n)의 분류를 위해 각 줄은 끝과 처음이 연결되어 있는 원형 리스트로 간주한다. 즉, 단어 w_1 의 왼쪽 단어는 w_n 이고, 단어 w_n 의 오른쪽 단어는 w_1 이다.

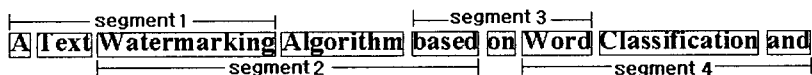
다음은 세그먼트 구성에 관한 설명이다. 줄의 첫 단어부터 시작하여 n 개의 연속된 단어들을 그룹화하고 이를 세그먼트(segment)라 하자. 그림 1은 $n=3$ 일 때의 세그먼트를 보여 준다. 줄의 첫 단어부터 시작하여 첫 번째 세그먼트를 만들고, 두 번째 세그먼트부터는 $n-1$ 개 단어를 건너 뛰어 세그먼트를 만든다. 그림 1에서는 네 개의 세그먼트가 만들어졌다. 한 세그먼트의 부류는 자신에 속한 n 개의 단어의 부류에 따라 결정된다. 그림 1은 단어가 2 부류인 경우로 하나의 세그먼트는 2^n 개로 분류되며, $n=3$ 이므로 8 부류가 된다. 이 예에서 네 개의 세그먼트는 111, 100, 011, 100 부류이다. 만일 표 2를 기준으로 단어를 4 부류 한다면 세그먼트는 총 4^n 개가 되고, $n=3$ 인 경우 $4^3=64$ 개로 분류된다.

2.3 신호 삼입과 추출

이렇게 세그먼트 단위로 신호를 삼입하는 이유는 Ω_i^l 과 Ω_i^r 의 조작을 쉽게 하기 위함이다. 이 방법에서는 Ω_i^l 과 Ω_i^r 이 단어 i 가 아닌 세그먼트 i 의 중앙 단어를 중심으로 왼쪽 단어와의 간격의 통계값, 오른쪽 단어와의 간격의 통계값을 나타낸다. 그림 1에서 볼 수 있듯이 하나의 세그먼트는 3개의 단어로 구성되므로 가운데 단어를 좌우로 이동하여 원하는 Ω_i^l 과 Ω_i^r 을 만들 수 있고, i 번째 세그먼트의 가운데 단어 이동은 $i+1$ 번째 세그먼트에 영향을 미치지 않는다.

각 세그먼트 부류에 삼입되는 신호의 양은 같다. 하나의 세그먼트에 p bits를 삼입한다고 할 때, 알고리즘에 의해 삼입되는 신호의 총량은 $L(\text{세그먼트 부류 수}) * p$ bits이다.

간단한 신호 삼입 규칙으로 다음과 같다.



$l(w_i)$	19	40	155	107	58	25	61	138	38
$C(w_i)$	1	1	1	0	0	1	1	0	0
segment class	111		100		011		100		

그림 1 단어 분류($K=2$)와 세크먼트 분류($n=3, 2^3=8$)

규칙 1: ($p=1, L=8, \Omega=\mu, \text{payload}=8 \text{ bits}$)

if($\mu \leq \mu_r$) signal 1,
otherwise signal 0.

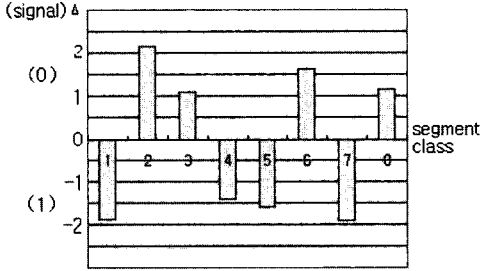


그림 2 신호 (10011010) 삽입 예

규칙 2: ($p=3, L=8, \Omega=\mu, \text{payload}=24 \text{ bits}$)

$\Delta = \mu - \mu_r$ 일 때,
if($3.5 < \Delta < 4.5$) signal 000(0),
else if($2.5 < \Delta < 3.5$) signal 001(1),
else if($1.5 < \Delta < 2.5$) signal 010(2),
else if($0.5 < \Delta < 1.5$) signal 011(3),
else if($-1.5 < \Delta < -0.5$) signal 100(4),
else if($-2.5 < \Delta < -1.5$) signal 0 101(5),
else if($-3.5 < \Delta < -2.5$) signal 0 110(6),
else if($-4.5 < \Delta < -3.5$) signal 0 111(7).

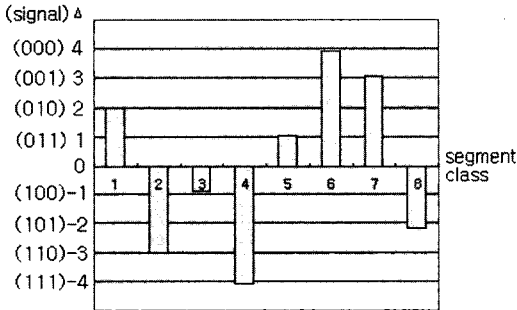


그림 3 신호 (010110100111011000001101) 삽입 예

규칙 1은 세그먼트 부류별 1 bit 삽입하는 것으로 총 8 bits의 신호를 삽입한다. 규칙 2는 Δ (단어의 왼쪽 공백 평균과 오른쪽 공백 평균의 차이)를 조절하여 신호를 삽입하는 방법으로, $\Delta > 0$ 은 왼쪽 공백의 평균이 오른쪽 공백의 크기보다 큰 것을 의미하며, $\Delta < 0$ 은 왼쪽 공백의 평균이 오른쪽 공백의 크기보다 작은 것을 의미한다. 그림 2는 규칙 1에 의해 세그먼트 부류별 1 bit를 삽입하는 예로 신호 '10011010'가 삽입되었다. 그림 3은 규칙 2에 의해 세그먼트 부류별 3 bits를 삽입하는 삽입 예이다. 세그먼트 부류 0에 신호 010 ($\Delta=2$), 세그먼트 부류 1에 신호 110 ($\Delta=-3$), 이런 방법으로 세그먼트 8 부류에 3 bits*8=24 bits가 삽입된다. Δ 의 구간을 조절함으로써 더 많은 bits를 삽입할 수 있다.

신호 추출은 다음과 같은 과정에 의해서 이루어진다.

1. 신호가 삽입된 영상을 입력받은 후 구조 분석 모듈 [18]을 이용하여 어절 단위의 분할을 한다.
2. 단어 부류와 세그먼트 부류로 분류한다.
3. 세그먼트내의 단어 사이의 간격을 측정한다.
4. 통계적 분포를 계산하고 인코딩 규칙에 따라 신호를 검출한다.

3. 실험 및 분석

이 장에서는 2.2절의 표 1과 표 2를 단어 분류 기준으로 사용하고, 2.3절의 규칙 1과 규칙 2를 신호 삽입 규칙으로 사용한 실험 예를 보인다. 또한 실험 결과를 통해 비인지성과 강인성을 분석하고 알고리즘의 특성에 의한 장점을 설명한다.

3.1 실험

먼저 세그먼트 부류의 발생 위치의 임의성에 대해 조사하였다. 그림 4에서는 세 개의 연속된 단어로 구성된 하나의 세그먼트가 직사각형으로 표시되어 있고, 그 직사각형의 가운데 위쪽에 세그먼트 부류를 표시하였다. 그림 4의 텍스트 블록에 대해 알아보기 쉽게 세그먼트 부류를 표로 제시하면 표 3과 같다.

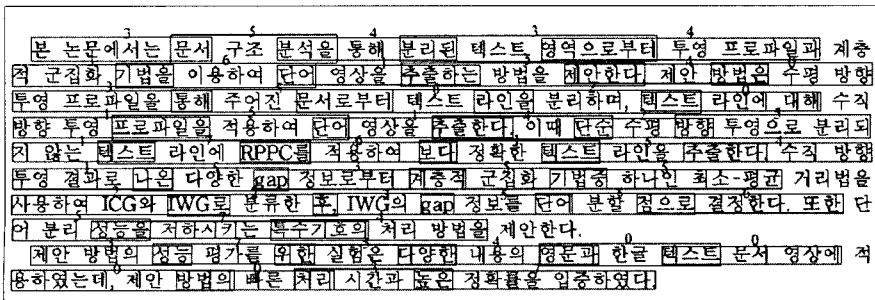


그림 4 세그먼트 부류의 발생 위치

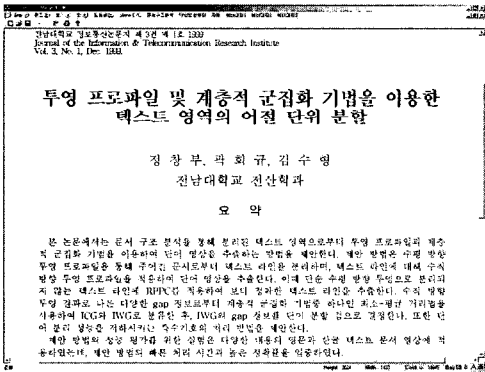
표 3 세그먼트 부류의 분포

줄 번호	세그먼트 수	세그먼트 부류
1	5	3, 5, 4, 3, 4
2	6	3, 6, 1, 5, 4, 0
3	5	3, 5, 6, 2, 0
4	6	1, 5, 1, 4, 2, 5
5	6	1, 7, 6, 1, 5, 4
6	5	1, 5, 3, 5, 5
7	6	2, 2, 3, 5, 5, 6
8	3	5, 7, 4
9	6	3, 7, 5, 4, 0, 0
10	4	0, 0, 3, 7

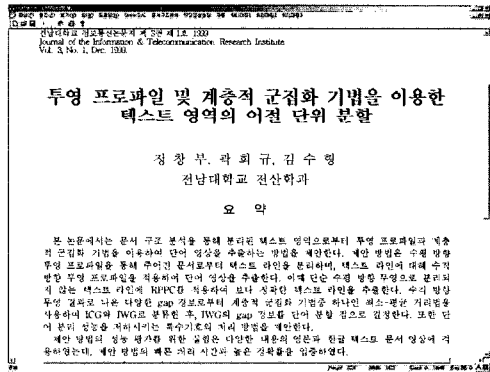
그림 5와 그림 6은 단어 2 분류, 세그먼트 부류는 n=3으로 하여 신호 삽입 규칙 1을 적용한 것이다. 그림 5(a)는 원 영상, 그림 5(b)는 신호(10011010) 삽입 후이며 그림 5(c)는 단어의 이동된 부분을 표시한 것이다. 그림 5(a)와 그림 5(b)를 비교해볼 때 육안으로 구별할 수 없어 비인지성을 만족한다. 그림 5(d)와 5(e)는 신호

삽입전의 Δ 와 삽입 후의 Δ 를 비교한 것이다. 5(d)와 5(e)의 세그먼트 부류 1, 3, 5, 8은 삽입전과 후의 Δ 가 동일하다. 즉, 신호 삽입 조건을 만족하는 세그먼트 부류(1, 3, 5, 8)는 단어 이동을 하지 않아도 되어 단어의 이동을 최소화 할 수 있으며, 이동해야 하는 경우에도 세그먼트 내의 가운데 단어만 이동시키므로 원본 영상의 배치를 깨지 않으므로 비인지성에서 우수하다고 할 수 있다. 그림 6은 다른 영상에 대한 실험으로 그림 6(b)에는 신호 '10101010'이 삽입되었다. 또한 그림 6(d)와 6(e)의 Δ 에 의해 세그먼트 부류 8에는 단어 이동이 없이도 신호가 삽입되었음을 알 수 있다.

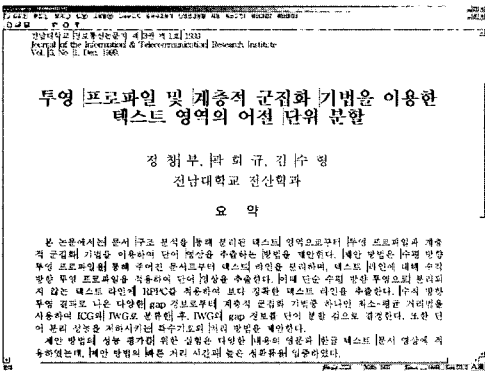
그림 7과 8은 단어 2 분류, n=3, 신호 삽입 규칙 2를 적용한 것이다. 그림 7(a)는 원 영상, 그림 7(b)은 신호(000 001 010 011 100 101 110 111)삽입 후이며 그림 7(c)은 단어의 이동된 부분을 표시한 것이다. 같은 영상으로 실험한 그림 5(c)와 비교해볼 때 7(c)의 이동 부분이 넓어진 것을 볼 수 있다. 신호 '000'과 신호 '111'에 해당하는 세그먼트 부류는 $\Delta=4$ 또는 $\Delta=-4$ 를 만족하



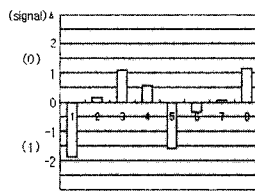
(a) 원 영상



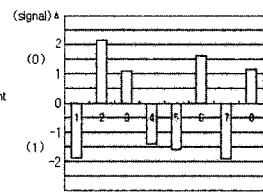
(b) 신호 (10011010) 삽입 후



(c) 이동된 부분 표시



(d) 신호 삽입 전



(e) 신호 삽입 후 Δ

그림 5 신호 삽입 예 1 (K=2, 신호 삽입 규칙 1, n=3)

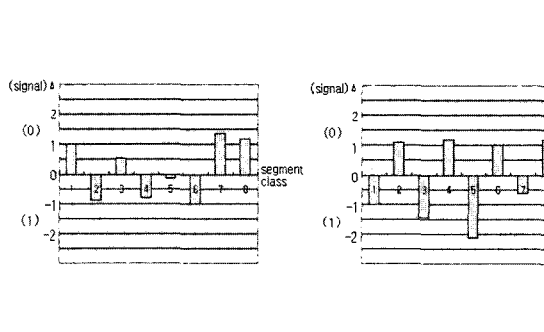
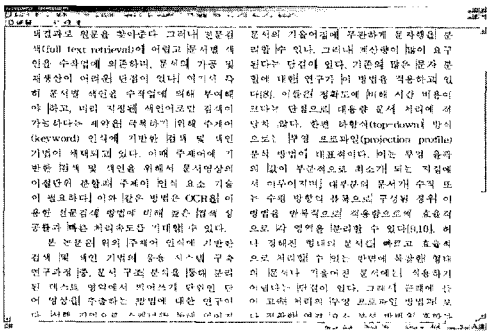
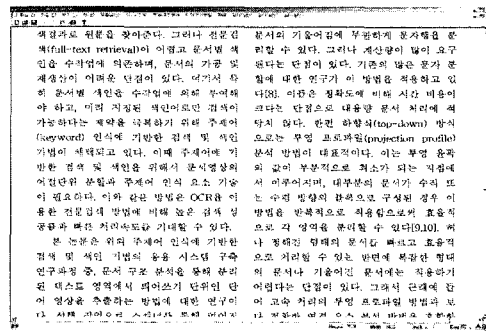
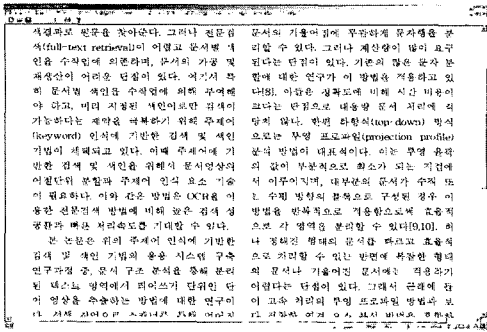


그림 6 신호 삽입 예 2 ($K=2$, 신호 삽입 규칙 1)

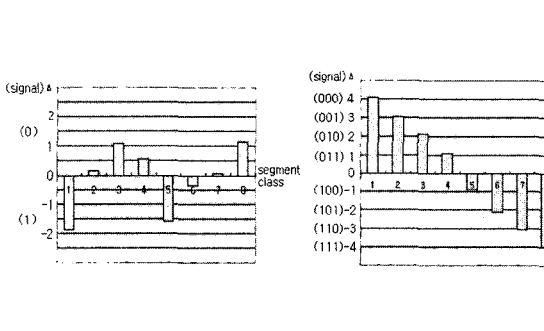
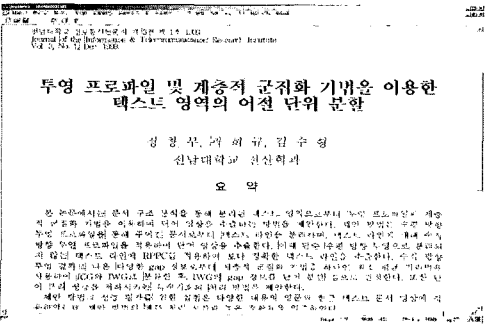
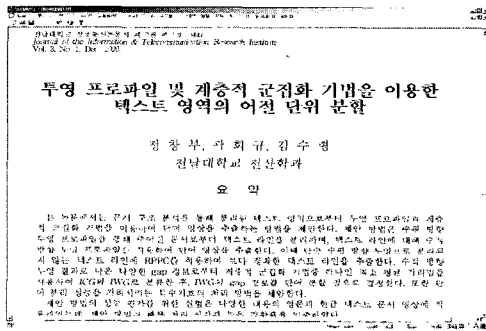
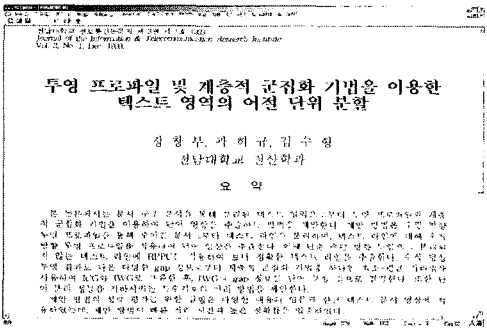
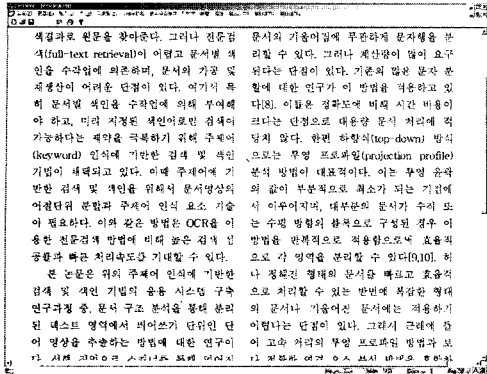
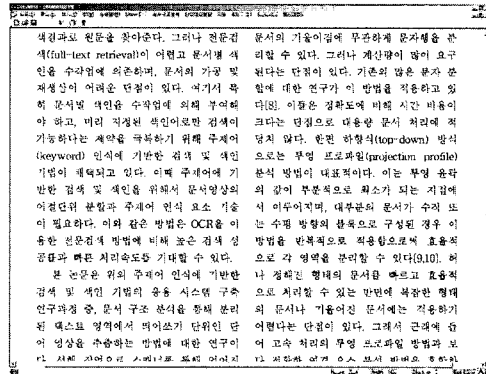


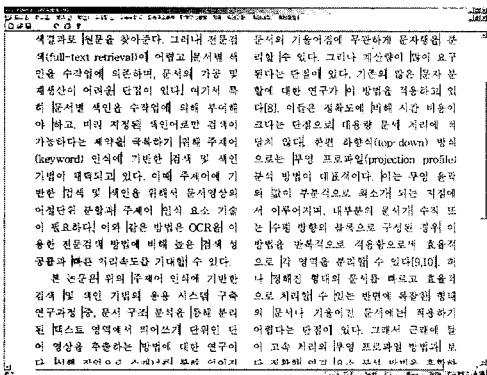
그림 7 신호 삽입 예 3 ($K=2$, 신호 삽입 규칙 2)



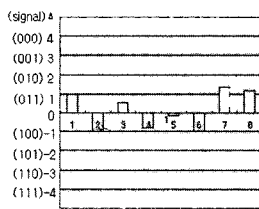
(a) 원 영상



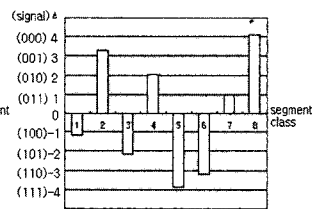
(b) 신호 삽입 후



(c) 이동된 부분 표시



(d) 신호 삽입 전 Δ



(e) 신호 삽입 후 Δ

그림 8 신호 삽입 예 4 ($K=2$, 신호 삽입 규칙 2)

도록 이동시켜야 하므로 그림 5(c)에 비해 단어의 이동량이 많기 때문이다. 그림 7(a)와 그림 7(b)을 비교해볼 때 이동된 부분이 육안으로 구별할 수 없어 비인지는 만족한다고 볼 수 있다. 그림 8은 다른 영상의 예로 신호 '100 001 101 010 111 110 011 000'이 삽입되었다. 많은 양의 텍스트 영상에 신호를 삽입하는 경우에는 단어 부류의 발생 빈도와 세그먼트 부류의 발생 빈도가 높아지므로 통계적인 의미의 Δ 는 신뢰성이 높다. 이런 경우에는 규칙 2를 적용하여 Δ 의 범위를 조절하여 더 많은 신호를 삽입할 수 있다. 예를 들어 Δ 의 신호 간격을 0.5로 하여 신호 16 구간을 만든다면 각 신호 구간은 4 bits로 표현되며 세그먼트 8 부류에 총 32 bits (4 bits*8)를 삽입 할 수 있다. 또한 Δ 의 전체 범위를 넓히거나 좁히는 방법으로 삽입되는 신호량을 조절할 수 있다.

그림 9는 영문 영상에 단어 4 부류, 신호 삽입 규칙 1을 적용한 것이다. 단어 4 부류의 부류 방법 (2.2 절의 표 4)은 하나의 단어의 부류를 결정하는데 자신을 포함하여 총 5개의 단어 (왼쪽 2개, 오른쪽 2개)가 필요하다.

marking scheme, one can invent a distortion (or a combination of distortions) that will prevent detection of the watermark while leaving the perceptual value of the previously watermarked object undiminished. We are not limited in this process to the distortions produced by common analogue equipment, or usually applied by end users with common image processing software. Moreover, the quality requirements of pirates are often lower than those of content owners who have to decide how much quality degradation to tolerate in return for extra protection offered by embedding a stronger signal.

(a) 그림 10의 부분 영상

marking scheme, one can invent a distortion (or a combination of distortions) that will prevent detection of the watermark while leaving the perceptual value of the previously watermarked object undiminished. We are not limited in this process to the distortions produced by common analogue equipment, or usually applied by end users with common image processing software. Moreover, the quality requirements of pirates are often lower than those of content owners who have to decide how much quality degradation to tolerate in return for extra protection offered by embedding a stronger signal.

(b) 신호(64 bits) 삽입 영상의 부분 영상
그림 9 영문 영상에 적용한 예(단어 4 부류, 신호 삽입 규칙 1)

watermark while leaving the perceptual value of the previ-
 watermark while leaving the perceptual value of the previ-

(a) 3번째 줄

analogue equipment, or usually applied by end users with
 analogue equipment, or usually applied by end users with

(b) 4번째 줄

그림 10 줄 단위 영상의 신호 삽입 전과 후

그러나 한글 텍스트 영상은 영문 영상에 비해 줄 단위 별로 단어의 수가 비교적 적게 분포하기 때문에, 하나의 단어의 부류를 결정하는데 5개 단어를 필요로 하는 단어 4 부류 방법을 적용하기에는 적합하지 않았다. 이러한 이유로 단어 4 부류 분류 방법은 영문 영상에만 적용하였다. 삽입되는 총 신호량은 64 bits ($1 \text{ bits} \times 4^3$)이다. 그림 9(a)는 원 영상 그림 11의 부분 영상이며 그림 9(b)는 신호(임의의 64 bits) 삽입 후이다. 비인지성 관찰을 위해 그림 10(a), (b)는 줄 단위로 신호 삽입 전과 후를 크게 확대한 것이다.

그림 12는 그림 11을 대상으로 신호 삽입을 하였을 때 단어들의 이동 크기 분포를 조사한 것이다. 단어를 왼쪽으로든 오른쪽으로든 1 픽셀 이동한 단어는 68개였으며, 2 픽셀 이동 단어는 37개, 3 픽셀 이동 단어는 7개, 4 픽셀 이동 단어는 3개 조사되었다. 그림 12를 통해 많이 이동된 단어의 수는 적음을 알 수 있다.

3.2 토론

본 절에서는 3.1절 실험을 통해 보았던 알고리즘의 특성을 설명하고, 이러한 특성에 의한 알고리즘의 장점을 설명한다.

1. 총 $p \times L$ bits의 신호를 삽입할 수 있다. L 은 이웃한 단어들의 조합으로 결정되어지므로 단어 부류(K)가 많아지면 L 의 부류도 많아진다. 그러나 K 가 커지면 상대적으로 부류별 빈도수 $|S(k)|$ 가 작아지므로 p (각 세그먼트에 삽입되는 신호양)가 따라서 작아지는 tradeoff 관계를 갖는다. 따라서 신호를 삽입할 텍스트 영상의 대상에 따라 신호 삽입 양을 조절할 수 있다.
2. 각 세그먼트가 텍스트 영상 전체에 임의로 발생하고 고르게 분포하기 때문에 삽입되는 신호 또한 고르게 분포하게 된다. 이러한 특징으로 인해 비인지성과 강인성면에서 장점을 갖는다.
3. 특정 세그먼트 부류의 통계값이 삽입할 신호 규칙과 이미 일치한다면 원본에 아무런 조작 없이도 신호 삽입이 가능하다. 이것은 비인지성 면에서 우수한 장점이다(실험 3.1 참조).

marking scheme, one can invent a distortion (or a combination of distortions) that will prevent detection of the watermark while leaving the perceptual value of the previously watermarked object undiminished. We are not limited in this process to the distortions produced by common analogue equipment, or usually applied by end users with common image processing software. Moreover, the quality requirements of pirates are often lower than those of content owners who have to decide how much quality degradation to tolerate in return for extra protection offered by embedding a stronger signal.

II. A BENCHMARK

Digital watermarking remains a largely untested field and very few authors have published extensive tests on their systems (e.g., [13]). A benchmark is needed to highlight promising areas of research by showing which techniques work better than others but also to be able to compare quickly new algorithms which appear in the literature. Until now most papers have used their own series of tests, their own pictures and their own methodology. So comparison is impossible without re-implementing the method and trying separately. But then, the implementation might be very different, and probably weaker, than the one of the original authors. With a common benchmark authors would just need to provide a brief table of results and other researchers would then have a fairly good idea of the performances of the proposed scheme and may make more thorough evaluation if interested.

For this benchmark we consider the watermarking process for embedding and recovering as a black box. For instance, some systems employ synchronisation templates or transformation invariants to survive some geometrical transformations. These template help to detect specific geometrical transformation. An inverse transformation is the applied to the image and the watermark is extracted for the modified image. The combination of the template and the embedding/extracting process form the watermarking algorithm as a whole, so they should not be evaluated together.

A. General procedure

For each image in a determined set, we used the following procedure for our tests.

1. Embed a mark with the strongest strength which does not introduce annoying effects. In other words, embed the mark such that the quality of the output, for a given quality metric, is greater than a given minima.
2. Apply a set of given distortions.
3. For each distorted image try to extract the watermark using a success/failure approach, that is to consider the extraction successful if and only if the payload is fully recovered without error.

Note that if the watermarking tools has a command line interface which allows to modify the parameters of the embedding and to check the error rate after extraction of the watermark, this procedure can be easily automated using Unix shell or Perl scripts.

그림 11 영문 원 영상

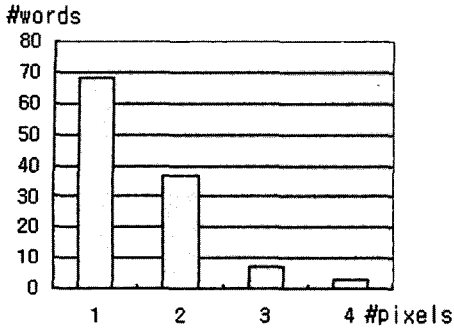


그림 12 단어들의 이동 크기 분포

4. 신호 추출 시 발생하는 오류는 줄 영역 또는 단어 영역 추출 등 구조 분석 처리 과정에서 발생한다. 기존의 텍스트 영상을 위한 워터마킹 방법들은 신호 삽입을 위해 일정한 픽셀만큼 이동하므로 신호의 강인성은 구조 분석기의 오류에 민감하다. 그러나 제안하는 방법은 세그먼트 내의 상대적인 거리를 이용하고, 신호가 영상에 걸쳐 고루 분포하기 때문에 구조 분석기 오류에 더 강인하다.
5. 각 부류가 통계적인 분포를 이용하므로 영상의 일부 분만 있어도 신호를 추출할 수 있는 장점을 갖는다. 예를 들어 손상된 텍스트 영상의 일부분에서 구조 분석기가 제대로 단어로 추출해 내지 못해도 신호를 올바르게 추출할 수 있는 특징을 가지고 있다.
6. 단어 분류는 상대적 길이의 비교에 의한 것이므로 절대적인 크기 값으로 분류하는 것보다 신뢰성이 높으며, 텍스트 영상의 종류나 문자의 폰트와 크기에 상관없이 일반적으로 적용 할 수 있다.

4. 결론 및 향후 연구

이 논문은 단어 분류에 기반한 텍스트 영상을 위한 워터마킹 알고리즘을 제안하였다. 단어 분류라는 새로운 방법을 제시하였으며, 세그먼트 부류별 단어의 왼쪽과 오른쪽 공백 크기의 통계값을 사용하여 신호를 삽입하고 추출하는 알고리즘을 제시하였다. 실험을 통해 알고리즘의 비가시성을 보았으며 통계값을 조작함으로써 가지는 강인성도 설명하였다.

향후 연구로는 단어 분류 방법과 세그먼트 결합 방법의 개선이 있다. 또한 post script 파일을 대상으로 알고리즘의 적용 범위를 확장하려 한다.

참고 문헌

- [1] I.J. Cox, M.L. Miller, and J.A. Bloom, *Digital Watermarking*, Morgan Kaufmann, 2002.
- [2] B.N. Schilit, M.N. Price, G. Golovchinsky, and C.C. Marshall, "As we may read: the reading appliance revolution," *IEEE Computers*, Vol.32, No.1, pp.65-73, January 1999.
- [3] *CACM*, Special issue on Digital Libraries, Vol.38, No.4, April 1995.
- [4] *IEEE Computers*, Special issue on e-government, Vol.34, No.2, February 2001.
- [5] Y. Liu, J. Mant, E. Wong, and S.H. Low, "Marking and detection of text documents using transform-domain techniques," *Proceedings of SPIE*, Vol.3657, pp.317-328, 1999.
- [6] J.T. Brassil, S. Low, and N.F. Maxemchuk, "Copyright protection for the electronic distribution of text documents," *Proceedings of IEEE*, Vol.87, No.7, pp.1181-1196, July 1999.
- [7] S. Low and N.F. Maxemchuk, "Performance comparison of two text marking methods," *IEEE J. Select. Areas Commun*, Vol.16, No.4, pp.561-572, May 1998.
- [8] S. Low, N.F. Maxemchuk, and A. Lapone, "Document identification for copyright protection using centroid detection," *IEEE Trans. Commun*, Vol.46, No.3, pp.372-383, March 1998.
- [9] Ding Huang and Hong Yan, "Interword distance changes represented by sine waves for watermarking text images," *IEEE Trans. Circuits and Systems for Video Technology*, Vol.11, No.12, pp.1237-1245, Dec 2001.
- [10] J.T. Brassil, S. Low, N.F. Maxemchuk, and L. O'Gorman, "Marking text features of document images to deter illicit dissemination," *Proceedings of 12th IAPR*, Vol.2, pp.315-319, 1994.
- [11] J.T. Brassil, S. Low, N.F. Maxemchuk, and L. O'Gorman, "Electronic marking and identification techniques to discourage document copying," *IEEE INFOCOM '94*, Vol.3, pp.1278-1287, 1994.
- [12] J.T. Brassil, S. Low, N.F. Maxemchuk, and L. O'Gorman, "Electronic marking and identification techniques to discourage document copying," *IEEE J. Select. Areas Commun*, Vol.13, No.8, pp.1495-1504, Oct 1995.
- [13] T. Amano and D. Misaki, "A feature calibration method for watermarking of document images," *Proceedings of ICDAR*, pp.91-94, 1999.
- [14] 김영원, 문경애, 오일석, "텍스트 문서 영상의 화소 수준 워터마킹 알고리즘", 제14회 영상 처리 및 이해에 관한 워크샵, 제주 롯데 호텔, pp.31-36, 2002.
- [15] A. Bhattacharjya and H. Ancin, "Data embedding in text for a copier system," *Proceedings of the ICIP*, Vol.2, pp.245-249, 1999.
- [16] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM System Journal*, Vol.35, pp.313-336, 1996.
- [17] <http://caelum.r.dl.itc.u-tokyo.ac.jp/~nakagawa/academic-result.html>
- [18] 박희규, 문서 영상의 단어 단위 분할 및 단어 영상의

속성 추출에 관한 연구, 전남대학교 박사 학위 논문, 2001.



김 영 원

2001년 전북대학교 컴퓨터과학과 졸업 (학사). 2003년 전북대학교 컴퓨터정보학과 졸업(석사). 2003년 3월~현재 전북대학교 컴퓨터정보학과 박사과정. 관심분야는 워터마킹, 문자인식, 컴퓨터비전



오 일 석

1984년 서울대학교 컴퓨터공학과 졸업 (학사). 1992년 KAIST 전산학과 졸업 (석사, 박사). 1992년 9월~현재 전북대학교 전자정보공학부 교수. 관심분야는 문서영상 처리, 패턴인식, 유전알고리즘의 패턴인식 응용