



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2015년04월24일

(11) 등록번호 10-1515040

(24) 등록일자 2015년04월20일

(51) 국제특허분류(Int. Cl.)

H04N 1/387 (2006.01) H04N 1/41 (2006.01)

(21) 출원번호 10-2014-0004332

(22) 출원일자 2014년01월14일

심사청구일자 2014년01월14일

(56) 선행기술조사문헌

KR1020090099394 A

KR1020020077209 A

KR1020070082098 A

JP2006208992 A

(73) 특허권자

조선대학교산학협력단

광주광역시 동구 필문대로 309 (서석동)

(72) 발명자

이정아

광주광역시 북구 삼정로 108, 102동 1702호 (두암동, 현대아파트)

마스우드 아프라크테

광주광역시 북구 월동로 12 무등빌 201호

(74) 대리인

특허법인아이엠

전체 청구항 수 : 총 13 항

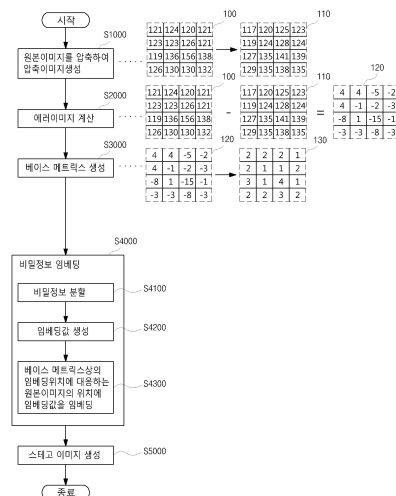
심사관 : 장석환

(54) 발명의 명칭 압축 이미지와 원본 이미지의 차이를 이용한 모듈식 스테고 이미지 생성방법

(57) 요약

본 발명은 스테고 이미지 생성방법에 관한 것으로, 보다 구체적으로는 원본 이미지와 압축 이미지의 차이인 에러 이미지로부터 임베딩할 비밀정보의 크기 및 위치에 관한 정보로 이루어지는 베이스 매트릭스를 계산하고, 계산된 베이스 매트릭스의 원소 값에 따라 비밀정보를 모듈식으로 임베딩하여 스테고 이미지를 생성할 수 있는 스테고 이미지 생성방법에 관한 것이다.

대 표 도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호 2013R1A1A3012335

부처명 미래창조과학부

연구관리전문기관 한국연구재단

연구사업명 여성과학자 일반연구지원사업

연구과제명 확장가능한 자가오류 탐지 및 오류회복 연산기 설계

기 여 율 1/1

주관기관 조선대학교

연구기간 2013.06.01 ~ 2014.05.31

명세서

청구범위

청구항 1

컴퓨터가 원본 이미지에 비밀정보를 임베딩하여 스테고 이미지(Stego-Images)를 생성하는 스테고 이미지 생성방법으로서,

특정한 이미지 품질 계수(IQF:image quality factor)에 따라 상기 원본 이미지를 압축하여 압축 이미지를 생성하는 단계;

상기 원본 이미지와 상기 압축 이미지의 픽셀 값 차이인 에러 이미지를 계산하는 단계;

상기 에러 이미지의 각 픽셀에 대해 비트 조합의 경우의 수가 픽셀 값을 넘지 않는 최대의 비트 수를 계산하고, 계산된 비트 수를 원소로 하는 행렬인 베이스 매트릭스(Base Matrix)를 생성하는 단계; 및

상기 베이스 매트릭스의 각 원소의 비트 수의 크기에 따라 상기 비밀정보를 임베딩할 위치를 결정하고 결정된 임베딩 위치들에 대응하는 상기 원본 이미지의 위치에 상기 비밀정보를 임베딩하여 상기 스테고 이미지를 생성하는 단계;를 포함하는 것을 특징으로 하는 스테고 이미지 생성방법.

청구항 2

제 1 항에 있어서,

상기 베이스 매트릭스는 아래의 수학적 식 1에 의해 계산되는 것을 특징으로 하는 스테고 이미지 생성방법.

[수학적 식 1]

$$Base_{i,j} = \lfloor \log_2(|ErrorImg_{i,j}|+1) \rfloor + OEF$$

여기서, $Base_{i,j}$ 는 상기 베이스 매트릭스의 i 행, j 열의 비트 수, $ErrorImg_{i,j}$ 는 상기 에러 이미지의 i 행, j 열의 픽셀 값, OEF(optimal extension field)는 최적 확장체 값이다.

청구항 3

제 2 항에 있어서,

상기 임베딩 위치는 상기 베이스 매트릭스에서 비트 수가 '2'이상인 값을 갖는 원소의 위치인 것을 특징으로 하는 스테고 이미지 생성방법.

청구항 4

제 3 항에 있어서,

상기 비밀정보를 임베딩하는 과정은,

상기 비밀정보를 특정한 길이의 비트 블록으로 분할하는 단계;

상기 비트 블록의 십진 값을 임베딩할 임베딩 값들로 변환하는 단계; 및

상기 임베딩 값들을 상기 임베딩 위치들에 순차적으로 하나씩 임베딩하는 단계;를 포함하는 것을 특징으로 하는 스테고 이미지 생성방법.

청구항 5

제 4 항에 있어서,

상기 임베딩 값들로 변환하는 단계는,

상기 십진 값을 상기 임베딩 위치들 중 최초 임베딩 위치의 비트 수로 나누어 계산된 나머지를 상기 최초 임베딩 위치의 임베딩 값으로 생성하고,

이전에 계산된 몫을 다음 임베딩 위치의 비트 수로 나누어 계산된 나머지를 상기 다음 임베딩 위치의 임베딩 값으로 생성하고,

순차적으로 이전에 계산된 몫을 다음 임베딩 위치의 비트 수로 나누어 나머지를 계산하여 임베딩 값을 생성하되, 피제수인 이전에 계산된 몫보다 제수인 비트 수가 커질 때까지 나눗셈을 수행하여 임베딩 값을 생성하는 것을 특징으로 하는 스테고 이미지 생성방법.

청구항 6

제 4 항에 있어서,

상기 임베딩하는 단계는,

해당 임베딩 위치의 임베딩 값이 '2'이상일 경우,

상기 임베딩 값에, 해당 임베딩 위치에 대응하는 원본 이미지의 픽셀값을 상기 베이스 매트릭스의 해당 비트 수로 나눈 나머지를 감한 제1 변환 값, 상기 제1 변환 값에 해당 비트 수를 더한 제2 변환 값 또는 상기 제1 변환 값에 해당 비트 수를 감한 제3 변환값을 원본 이미지의 해당 픽셀값에 더하여 스테고 픽셀로 변환함으로써 수행되는 것을 특징으로 하는 스테고 이미지 생성방법.

청구항 7

제 4 항에 있어서,

상기 임베딩하는 단계는,

해당 임베딩 위치의 임베딩 값이 '2'이상일 경우,

해당 임베딩 위치의 원본 이미지 픽셀 값에 상기 임베딩 값을 빼고, 상기 베이스 매트릭스의 해당 비트 수로 나누어, 나눈 몫보다 크거나 같은 정수 중 가장 작은 정수에 상기 베이스 매트릭스의 해당 비트수로 곱하고, 상기 임베딩 값을 더하여 스테고 픽셀로 변환함으로써 수행되는 것을 특징으로 하는 스테고 이미지 생성방법.

청구항 8

제 6 항에 있어서,

상기 임베딩하는 단계는,

해당 임베딩 위치의 임베딩 값이 '2'미만일 경우,

최하위 비트 매칭 기법(LSB matching)을 이용하여 원본 이미지의 해당 픽셀값을 스테고 픽셀로 변환함으로써 수행되는 것을 특징으로 하는 스테고 이미지 생성방법.

청구항 9

제 8 항에 있어서,

상기 최하위 비트 매칭 기법은

임베딩 값이 '2'미만인 한 쌍의 임베딩 값(제1 임베딩 값 및 제2 임베딩 값으로 이루어짐)과 해당 임베딩 값이 임베딩될 한 쌍의 원본 이미지 픽셀 값(제1 픽셀 값, 제2 픽셀 값으로 이루어짐)을 읽어들이는 단계;

상기 제1 임베딩 값이 상기 제1 픽셀 값의 최하위 비트 값과 동일할 경우,

만약, 상기 제1 픽셀 값을 '2'로 나누고, 나눈 값을 넘지 않는 최소 정수 값에 상기 제2 픽셀 값을 더한 값의 최하위 비트 값이 상기 제2 임베딩 값과 동일하지 않다면, 상기 제2 픽셀은 상기 제2 픽셀의 픽셀 값에 '1'을 더하거나 뺀 값의 스테고 픽셀로 변환되고, 그 외의 경우에는 상기 제2 픽셀은 상기 제2 픽셀의 동일한 값과 동일한 값의 스테고 픽셀로 변환되며, 상기 제1 픽셀은 상기 제1 픽셀과 동일한 값의 스테고 픽셀로 변환되며,

상기 제1 임베딩 값이 상기 제1 픽셀 값의 최하위 비트 값과 동일하지 않을 경우,

만약, 상기 제1 픽셀 값에 '1'을 뺀 값을 '2'로 나누고, 나눈 값을 넘지 않는 최소 정수 값에 상기 제2 픽셀 값을 더한 값의 최하위 비트 값이 상기 제2 임베딩 값과 동일하다면, 상기 제1 픽셀은 상기 제1 픽셀의 픽셀 값에 '1'을 뺀 값의 스테고 픽셀로 변환되고, 그 외의 경우 상기 제1 픽셀은 상기 제2 픽셀과 동일한 값의 스테고 픽셀로 변환되며, 상기 제2 픽셀은 상기 제2 픽셀과 동일한 값의 스테고 픽셀로 변환되는 단계;를 포함하는 것을 특징으로 하는 스테고 이미지 생성방법.

청구항 10

제 2 항 내지 제 9 항 중 어느 한 항에 있어서,

상기 스테고 이미지를 생성하는 단계 이후에,

상기 스테고 이미지, 상기 이미지 품질 계수 및 상기 최적 확장체 값을 통신망을 통해 외부의 수신처로 송신하는 단계를 더 포함하는 것을 특징으로 하는 스테고 이미지 생성방법.

청구항 11

제 2 항 내지 제 9 항 중 어느 한 항의 스테고 이미지 생성방법을 수행하는 스테고 이미지 생성 프로그램이 저장된 컴퓨터로 읽을 수 있는 저장매체.

청구항 12

제 2 항 내지 제 9 항 중 어느 한 항의 스테고 이미지 생성방법을 수행하는 스테고 이미지 생성 프로그램이 저장되고 통신망을 통해 상기 스테고 이미지 생성 프로그램을 전송할 수 있는 컴퓨터 장치.

청구항 13

제 2 항 내지 제 9 항 중 어느 한 항의 스테고 이미지 생성방법을 수행하는 스테고 이미지 생성 프로그램이 탑재된 영상 처리 장치.

발명의 설명

기술 분야

본 발명은 스테고 이미지 생성방법에 관한 것으로, 보다 구체적으로는 원본 이미지와 압축 이미지의 차이인 에러 이미지로부터 임베딩할 비밀정보의 크기 및 위치에 관한 정보로 이루어지는 베이스 매트릭스를 계산하고, 계산된 베이스 매트릭스의 원소 값에 따라 비밀정보를 모듈식으로 임베딩하여 스테고 이미지를 생성할 수 있는 스테고 이미지 생성방법에 관한 것이다.

[0001]

배경 기술

- [0002] 스테가노그래피(steganography)란 비밀정보가 전송되고 있다는 사실을 숨기는 기술로, 이미지나 비디오, 오디오와 같은 다양한 유형의 디지털 원본 미디어에 비밀정보를 숨겨 전송하는 기법을 뜻한다.
- [0003] 스테가노그래피 기법을 설계하는데 있어 고려되어야 하는 두 가지 요소는 얼마나 많은 양의 비밀정보를 임베딩할 수 있는가 하는 임베딩 비율과 임베딩된 비밀정보의 검출이 얼마나 어려운가 하는 검출 불가능성이 있다.
- [0004] 또한, 임베딩 비율과 검출 불가능성은 서로 절충적인 관계이며, 일반적으로 임베딩 비율이 높을수록 검출 불가능성은 낮아진다.
- [0005] 일부 종래의 스테가노그래피 기술은 임베딩 비율에 큰 관심을 갖고 설계되었으나 최근 검출 불가능성을 높이고자 하는 다양한 스테가노그래피 기법이 제안되고 있다.
- [0006] 간단한 최하위 비트 대체 기법(Simple LSB replacement)[1]은 픽셀의 최하위 비트에 비밀정보를 임베딩하여 원본 이미지를 직접 대체한다. 이러한 기법은 일부 원치 않는 통계 산물이 추가되는 것으로 인한 비밀 비트의 존재가 노출될 수 있다. 이러한 기법은 카이 스퀘어(chi-square) 공격[2]에 대해 비밀정보가 쉽게 검출된다.
- [0007] 최하위 비트 매칭(LSBM:LSB matching) 기법[3]은 최하위 비트의 대체 후에 수도 랜덤 넘버 제너레이터(PRNG:pseudo random number generator)에 의해 최하위 비트를 무작위로 변화시키는 기법이고, 수정된 최하위 비트 매칭(LSB matching revisited) 기법[4]은 최하위 비트의 픽셀 값을 무작위로 증가시키거나 감소시키지 않는 최하위 비트 매칭 기법의 수정된 방법이다.
- [0008] 최하위 비트 매칭 기법과 수정된 최하위 비트 매칭 기법은 모두 카이 스퀘어 공격에 의해 검출이 불가능하다.
- [0009] 에지 적응적(EA:Edge adaptive) 이미지 스테가노그래피 기법[5]은 수정된 최하위 비트 매칭 기법에 기반하여 비밀 정보를 임베딩하는 기법으로 평활영역(smooth area)은 그대로 유지하고 가능한 한 에지 영역에서부터 임베딩을 시작한다. 이 방법은 최대 임베딩 비율이 1[bpp](bit per pixel)로 제한되지만, 스테고 이미지의 시각 품질과 보안이 양호한 것으로 증명되고 있다.
- [0010] 또 다른 기법으로 HUGO기법[6]은 고차원 이미지 모델을 사용하는 기법으로, 이 방법은 2차 감산 픽셀 인접성 모델(SPAM:subtractive pixel adjacency model)[7]의 특징을 이용하여 서포트 벡터 머신(SVM:training support vector machine) 기반 스테가노그래피 분석기를 훈련시켜 평가된다.
- [0011] 마지막으로 앙상블 분류기(ensemble classifiers)[8]는 시간 및 정확성 측면에서 서포트 벡터 머신 기반의 스테가노 분석기에 비해 양호한 성능을 갖는다.
- [0012] 반면에, 고전적인 스테가노그래피 기법은 더 큰 임베딩 비율을 제공하는데 이는 대개 1[bpp]를 초과한다.
- [0013] 픽셀 값 차별화(PVD:pixel value differencing) 기법[9]은 높은 임베딩 비율에서 높은 감지 불가능성을 갖는데 원본 이미지를 두 개의 접속 픽셀을 포함하는 비중첩 블록으로 분할하고 비밀 정보 임베딩을 위해 각각의 블록 내에서 픽셀 차이를 수정하는 방법이다. 또한, 수정된 픽셀 값 차별화(PVD) 기법[10]은 픽셀 값 차별화 기법[9]보다 나은 보안성을 갖는다.
- [0014] 또 다른 방법[11]으로 인접 픽셀들 사이의 상관도를 이용하여 보다 큰 신호대잡음비(PSNR)값과 큰 임베딩 용량을 허용하는 기법이 있다.
- [0015] 비트 평면 복잡도 분할(BPCS:Bit-plane complexity segmentation) 기법[12]은 비트 공간의 복잡도를 계산하여 높은 복잡도의 영역에 임베딩하는 아이디어를 적용하여 스테고 이미지를 생성한다.
- [0016] 제키(Zeki) 등[13]은 호스트 이미지의 최소 열화를 달성하는 방식을 제안하였고, 파라(Parah) 등[14]은 비밀 정보를 가변 길이로 분할하여 임베딩하는 기법을 제안하였다.
- [0017] 다중 베이스 기록 시스템(MBNS:multiple-base notational system) 기법[15]은 비밀정보를 동일한 크기의 블록으로 분할하여 모듈러 함수를 통해 원본 이미지를 변경하였고, 창(Chang) 등[16]은 연속 길이(run-length) 부호화 및 모듈러 연산을 통합하는 두 개의 비밀정보 은닉방안을 제안하였다.
- [0018] 솔레이마니(Soleimani) 및 니아찌토르시즈(NiaziTorshiz)[17]에 의한 또 다른 방안은 k-최하위 비트(K-LSB) 임베딩[18]보다 양호한 품질의 스테고 이미지를 생성하며, 픽셀 값 차별화 기법[9]과 같이 중첩하지 않는 2 픽셀 블록을 나누는 것으로부터 시작하여 다양한 모듈러 동작을 통해 비밀정보를 임베딩한다.

- [0019] 씨엔(Thien) 및 린(Lin)[19]은 모듈러 동작에 기반하여 512x512 호스트 이미지 내에 4[bpp]를 은닉하는 LSB 기반 방안을 제안하였다
- [0020] 첸(Chen)[20]은 무손실 비밀 데이터 압축이 적용되는 또 다른 모듈 기반 LSB 치환 방법을 논의하였다
- [0021] 또한, 이산 코사인 변환 기반의 임베딩 기법 및 LSB기반 접근 방법[21]을 통한 스테가노그래피 기법이 있다.

선행기술문헌

비특허문헌

- [0022] (비특허문헌 0001) [1] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, (1996). Techniques for data hiding. IBM systems journal, 35(3.4), 313-336.
- (비특허문헌 0002) [2] A. Westfeld, and A. Pfitzmann. "Attacks on steganographic systems." Information Hiding. Springer Berlin Heidelberg, 2000.
- (비특허문헌 0003) [3] A.D. Ker, "Improved detection of LSB steganography in grayscale images." Information Hiding. Springer Berlin Heidelberg, 2005.
- (비특허문헌 0004) [4] J. Mielikainen, "LSB matching revisited." Signal Processing Letters, IEEE 13.5 (2006): 285-287.
- (비특허문헌 0005) [5] W. Luo, F. Huang, and J. Huang. "Edge adaptive image steganography based on LSB matching revisited." Information Forensics and Security, IEEE Transactions on 5.2 (2010): 201-214.
- (비특허문헌 0006)
- (비특허문헌 0007)
- (비특허문헌 0008)
- (비특허문헌 0009) [9] X. Li , B. Li , X. Luo , B. Yang, and R. Zhu, (2013). Steganalysis of a PVD-based content adaptive image steganography. Signal Processing.
- (비특허문헌 0010) [10] X. Zhang , and S. Wang, (2004). Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security. Pattern Recognition Letters, 25(3), 331-339.
- (비특허문헌 0011) [11] C. C. Chang, and H. W. Tseng, (2004). A steganographic method for digital images using side match. Pattern Recognition Letters, 25(12), 1431-1437.
- (비특허문헌 0012) [12] H. Noda , J. Spaulding , M. N. Shirazi, and E. Kawaguchi, (2002). Application of bit-plane decomposition steganography to JPEG2000 encoded images. Signal Processing Letters, IEEE, 9(12), 410-413.
- (비특허문헌 0013) [13] A. M. Zeki , A. Abdul Manaf, and S. S. Mahmood, (2011). High watermarking capacity based on spatial domain technique. Information technology journal,10(7), 1367-1373.
- (비특허문헌 0014) [14] S. A. Parah, J. A. Sheikh, and G. M. Bhat, High Capacity Data Embedding using joint Intermediate Significant Bit (ISB) and Least Significant Bit (LSB) Technique, Journal of Information Engineering and Applications, ISSN 2224-5782 (print) ISSN 2225-0506 (online) Vol. 2, No.11, 2012.
- (비특허문헌 0015)
- (비특허문헌 0016)
- (비특허문헌 0017) [17] S.R. Soleimani, and M. Niazi Torshiz, A New High Quality Vision Non-Adaptive Steganographic Method, Using Module and Combined Functions, International Journal of Emerging Trends

in Signal Processing ISSN(Online)2319-9784, Volume 1 ,Issue 2, January 2013 .

(비특허문헌 0018) [18] C.K. Chan, and L. M. Cheng. Hiding data in images by simple lsb substitution. Pattern Recognition, 37:469?474, 2004.

(비특허문헌 0019)

(비특허문헌 0020) [20] S. K. Chen, (2011). A module-based LSB substitution method with lossless secret data compression. Computer Standards & Interfaces, 33(4), 367-371.

(비특허문헌 0021) [21] A. Cheddad, J. Condell, K. Curran, & P.M. Kevitt, 2010. Digital image steganography: Survey and analysis of current methods. Signal Processing, 90(3): 727-752.

발명의 내용

해결하려는 과제

[0023] 본 발명의 목적은 많은 양의 비밀정보를 임베딩하더라도 잡음이 적고 검출 불가능성은 높은 스테고 이미지 생성 방법을 제공하는 것이다.

[0024] 또한, 본 발명의 목적은 원본 이미지와 특정 이미지 품질 계수에 의해 압축된 압축 이미지의 차이인 에러 이미지를 이용하여 비밀정보를 임베딩할 위치 및 용량을 결정하고, 각 임베딩 위치에 모듈식으로 비밀정보를 임베딩하는 신규한 스테고 이미지 생성방법을 제공하는 것이다.

[0025] 본 발명의 목적들은 이상에서 언급한 목적들로 제한되지 않으며, 언급되지 않은 또 다른 목적들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0026] 상기의 목적을 달성하기 위하여 본 발명은 컴퓨터가 원본 이미지에 비밀정보를 임베딩하여 스테고 이미지 (Stego-Images)를 생성하는 스테고 이미지 생성방법으로서, 특정한 이미지 품질 계수(IQF:image quality factor)에 따라 상기 원본 이미지를 압축하여 압축 이미지를 생성하는 단계; 상기 원본 이미지와 상기 압축 이미지의 픽셀 값 차이인 에러 이미지를 계산하는 단계; 상기 에러 이미지의 각 픽셀에 대해 비트 조합의 경우의 수가 픽셀 값을 넘지 않는 최대의 비트 수를 원소로 하는 행렬인 베이스 매트릭스(Base Matrix)를 생성하는 단계; 및 상기 베이스 매트릭스의 각 원소의 비트 수의 크기에 따라 상기 비밀정보를 임베딩할 위치를 결정하고 결정된 임베딩 위치들에 대응하는 상기 원본 이미지의 위치에 상기 비밀정보를 임베딩하여 상기 스테고 이미지를 생성하는 단계;를 포함하는 것을 특징으로 하는 스테고 이미지 생성방법을 제공한다.

[0027] 바람직한 실시예에 있어서, 상기 베이스 매트릭스는 아래의 수학식 1에 의해 계산된다.

[0028] [수학식 1]

$$Base_{ij} = \lfloor \log_2(|ErrorImg_{ij}|+1) \rfloor + OEF$$

[0029] 여기서, $Base_{ij}$ 는 상기 베이스 매트릭스의 i행, j열의 비트 수, $ErrorImg_{ij}$ 는 상기 에러 이미지의 i행, j열의 픽셀 값, OEF(optimal extension field)는 최적 확장계 값이다.

[0031] 바람직한 실시예에 있어서, 상기 임베딩 위치는 상기 베이스 매트릭스에서 비트 수가 '2'이상인 값을 갖는 원소의 위치이다.

[0032] 바람직한 실시예에 있어서, 상기 비밀정보를 임베딩하는 과정은, 상기 비밀정보를 특정한 길이의 비트 블록으로 분할하는 단계; 상기 비트 블록의 십진 값을 임베딩할 임베딩 값들로 변환하는 단계; 및 상기 임베딩 값들을 상기 임베딩 위치들에 순차적으로 하나씩 임베딩하는 단계;를 포함한다.

- [0033] 바람직한 실시예에 있어서, 상기 임베딩 값들로 변환하는 단계는, 상기 십진 값을 상기 임베딩 위치들 중 최초 임베딩 위치의 비트 수로 나누어 계산된 나머지를 상기 최초 임베딩 위치의 임베딩 값으로 생성하고, 이전에 계산된 몫을 다음 임베딩 위치의 비트 수로 나누어 계산된 나머지를 상기 다음 임베딩 위치의 임베딩 값으로 생성하고, 순차적으로 이전에 계산된 몫을 다음 임베딩 위치의 비트 수로 나누어 나머지를 계산하여 임베딩 값을 생성하되, 피제수인 이전에 계산된 몫보다 제수인 비트 수가 커질 때까지만 나눗셈을 수행하여 임베딩 값을 생성한다.
- [0034] 바람직한 실시예에 있어서, 상기 임베딩하는 단계는, 해당 임베딩 위치의 임베딩 값이 '2'이상일 경우, 상기 임베딩 값에, 해당 임베딩 위치에 대응하는 원본 이미지의 픽셀값을 상기 베이스 메트릭스의 해당 비트 수로 나눈 나머지를 감한 제1 변환 값, 상기 제1 변환 값에 해당 비트 수를 더한 제2 변환 값 또는 상기 제1 변환 값에 해당 비트 수를 감한 제3 변환값을 원본 이미지의 해당 픽셀값에 더하여 스테고 픽셀로 변환함으로써 수행된다.
- [0035] 바람직한 실시예에 있어서, 상기 임베딩하는 단계는, 해당 임베딩 위치의 임베딩 값이 '2'이상일 경우, 해당 임베딩 위치의 원본 이미지 픽셀 값에 상기 임베딩 값을 빼고, 상기 베이스 메트릭스의 해당 비트 수로 나누어, 나눈 몫보다 크거나 같은 정수 중 가장 작은 정수에 상기 베이스 메트릭스의 해당 비트수로 곱하고, 상기 임베딩 값을 더하여 스테고 픽셀로 변환함으로써 수행될 수 있다.
- [0036] 바람직한 실시예에 있어서, 해당 임베딩 위치의 임베딩 값이 '2'미만일 경우, 최하위 비트 매칭 기법(LSB matching)을 이용하여 원본 이미지의 해당 픽셀값을 스테고 픽셀로 변환함으로써 수행된다.
- [0037] 바람직한 실시예에 있어서, 상기 최하위 비트 매칭 기법은 임베딩 값이 '2'미만인 한 쌍의 임베딩 값(제1 임베딩 값 및 제2 임베딩 값으로 이루어짐)과 해당 임베딩 값이 임베딩될 한 쌍의 원본 이미지 픽셀 값(제1 픽셀 값, 제2 픽셀 값으로 이루어짐)을 읽어들이는 단계; 상기 제1 임베딩 값이 상기 제1 픽셀 값의 최하위 비트 값과 동일할 경우, 만약, 상기 제1 픽셀 값을 '2'로 나누고, 나눈 값을 넘지 않는 최소 정수 값에 상기 제2 픽셀 값을 더한 값의 최하위 비트 값이 상기 제2 임베딩 값과 동일하지 않다면, 상기 제2 픽셀은 상기 제2 픽셀의 픽셀 값에 '1'을 더하거나 뺀 값의 스테고 픽셀로 변환되고, 그 외의 경우에는 상기 제2 픽셀은 상기 제2 픽셀의 동일한 값과 동일한 값의 스테고 픽셀로 변환되며, 상기 제1 픽셀은 상기 제1 픽셀과 동일한 값의 스테고 픽셀로 변환되며, 상기 제1 임베딩 값이 상기 제1 픽셀 값의 최하위 비트 값과 동일하지 않을 경우, 만약, 상기 제1 픽셀 값에 '1'을 뺀 값을 '2'로 나누고, 나눈 값을 넘지 않는 최소 정수 값에 상기 제2 픽셀 값을 더한 값의 최하위 비트 값이 상기 제2 임베딩 값과 동일하다면, 상기 제1 픽셀은 상기 제1 픽셀의 픽셀 값에 '1'을 뺀 값의 스테고 픽셀로 변환되고, 그 외의 경우 상기 제1 픽셀은 상기 제2 픽셀과 동일한 값의 스테고 픽셀로 변환되며, 상기 제2 픽셀은 상기 제2 픽셀과 동일한 값의 스테고 픽셀로 변환되는 단계;를 포함한다.
- [0038] 바람직한 실시예에 있어서, 상기 스테고 이미지를 생성하는 단계 이후에, 상기 스테고 이미지, 상기 이미지 품질 계수 및 상기 최적 확장체 값을 통신망을 통해 외부의 수신처로 송신하는 단계를 더 포함한다.
- [0039] 또한, 본 발명은 상기 스테고 이미지 생성방법을 수행하는 스테고 이미지 생성 프로그램이 저장된 컴퓨터로 읽을 수 있는 저장매체를 더 제공한다.
- [0040] 또한, 본 발명은 상기 스테고 이미지 생성방법을 수행하는 스테고 이미지 생성 프로그램이 저장되고 통신망을 통해 상기 스테고 이미지 생성 프로그램을 전송할 수 있는 컴퓨터 장치를 더 제공한다.
- [0041] 또한, 본 발명은 상기 스테고 이미지 생성방법을 수행하는 스테고 이미지 생성 프로그램이 탑재된 영상 처리 장치를 더 제공한다.

발명의 효과

- [0042] 본 발명은 다음과 같은 우수한 효과를 가진다.
- [0043] 먼저, 본 발명의 스테고 이미지 생성방법에 의하면, 많은 양의 비밀정보를 임베딩하더라도 신호대 잡음비(PSNR)가 높고, 비밀정보의 검출 불가능성은 낮은 장점이 있다.
- [0044] 또한, 본 발명의 스테고 이미지 생성방법에 의하면, 원본 이미지와 특정 이미지 품질 계수에 의해 압축된 압축 이미지의 차이인 에러 이미지로부터 베이스 메트릭스를 계산하고 베이스 메트릭스의 비트 수에 따라 비밀정보를 모듈식으로 임베딩할 수 있는 장점이 있다.

도면의 간단한 설명

- [0045] 도 1은 본 발명의 일 실시예에 따른 스테고 이미지 생성방법의 흐름도,
 도 2는 본 발명의 일 실시예에 따른 스테고 이미지 생성방법의 비밀정보를 설명하기 위한 도면,
 도 3은 본 발명의 일 실시예에 따른 스테고 이미지 생성방법의 임베딩 값을 생성하는 과정을 보여주는 도면,
 도 4는 본 발명의 일 실시예에 따른 스테고 이미지 생성방법의 임베딩 값을 생성하는 일례를 보여주는 도면,
 도 5는 본 발명의 일 실시예에 따른 스테고 이미지 생성방법에서 임베딩 값을 임베딩하는 과정을 설명하기 위한 도면,
 도 6은 씨엔 및 린의 기법[19]에 의해 생성된 스테고 이미지의 카이 스퀘어 공격[2]에 의한 비밀정보 검출가능성을 보여주는 그래프이고,
 도 7은 본 발명의 일 실시예에 따른 스테고 이미지 생성방법에 의해 생성된 스테고 이미지의 카이 스퀘어 공격[2]에 의한 비밀정보 검출가능성을 보여주는 그래프이다.

발명을 실시하기 위한 구체적인 내용

- [0046] 본 발명에서 사용되는 용어는 가능한 현재 널리 사용되는 일반적인 용어를 선택하였으나, 특정한 경우는 출원인이 임의로 선정한 용어도 있는데 이 경우에는 단순한 용어의 명칭이 아닌 발명의 상세한 설명 부분에 기재되거나 사용된 의미를 고려하여 그 의미가 파악되어야 할 것이다.
- [0047] 이하, 첨부한 도면에 도시된 바람직한 실시예들을 참조하여 본 발명의 기술적 구성을 상세하게 설명한다.
- [0048] 그러나, 본 발명은 여기서 설명되는 실시예에 한정되지 않고 다른 형태로 구체화 될 수도 있다. 명세서 전체에 걸쳐 동일한 참조번호는 동일한 구성요소를 나타낸다.
- [0049] 도 1은 본 발명의 일 실시예에 따른 스테고 이미지 생성방법의 흐름도이다.
- [0050] 도 1을 참조하면, 본 발명의 스테고 이미지 생성방법은 원본 이미지(100)의 특정 픽셀 값을 비밀정보가 임베딩된 스테고 픽셀로 변환하여 스테고 이미지를 생성하는 방법으로써, 실질적으로 스테고 이미지 생성 프로그램이 저장된 컴퓨터에 의해 수행된다.
- [0051] 또한, 상기 스테고 이미지 생성 프로그램은 컴퓨터를 수단으로 기능시켜 상기 스테고 이미지 생성방법의 각 단계를 수행함으로써 스테고 이미지를 생성한다.
- [0052] 또한, 상기 스테고 이미지 생성 프로그램은 HDD, CD, USB저장소, SD카드 등과 같은 저장장치에 저장되어 상기 컴퓨터에 의해 직접 읽혀짐으로써 본 발명의 스테고 이미지 생성방법이 수행되게 할 수 있고, 상기 스테고 이미지 생성 프로그램이 저장된 컴퓨터 장치로부터 통신망을 통해 상기 스테고 이미지 생성 프로그램을 전송받아 상기 컴퓨터가 본 발명의 스테고 이미지 생성방법을 수행할 수 있다.
- [0053] 또한, 상기 컴퓨터 장치는 상기 스테고 이미지 생성 프로그램을 저장하는 저장소, 통신망을 통해 데이터를 송수신할 수 있는 데이터 송수신 장치 및 상기 데이터 송수신 장치를 통해 상기 스테고 이미지 생성프로그램을 전송할 수 있는 중앙처리장치를 포함하는 서버 컴퓨터일 수 있다.
- [0054] 또한, 상기 컴퓨터는 퍼스널 컴퓨터뿐만 아니라 스마트 폰, 태블릿 피시 등 이미지 처리를 수행할 수 있는 다양한 컴퓨팅 장치일 수 있으며, GPU가 포함된 다양한 영상 처리 장치일 수 있다.
- [0055] 본 발명의 일 실시예에 따른 스테고 이미지 생성방법은 먼저, 특정한 이미지 품질 계수(IQF:image quality factor)에 따라 원본 이미지(100)를 압축하여 압축 이미지(110)를 생성한다(S1000).
- [0056] 또한, 상기 이미지 품질 계수는 이미지의 압축 정도를 결정하기 위한 계수으로써 상기 이미지 품질 계수가 작을수록 압축 정도는 커지며, 상기 압축 이미지(110)에 노이즈는 증가한다.

- [0057] 또한, 상기 압축 이미지(100)를 생성하는 방법은 종래의 다양한 이미지 압축방법을 이용할 수 있으며, 예를 들면, JPEG 표준압축방법으로 압축할 수 있다.
- [0058] 다만, 압축 정도는 상기 이미지 품질 계수에 따른다.
- [0059] 즉, 상기 이미지 품질 계수는 원본 이미지의 크기나 화질에 따라 사용자 설정할 수 있는 값이다.
- [0060] 다음, 상기 원본 이미지(100)와 상기 압축 이미지(110)의 픽셀 값의 차이를 계산하여 에러 이미지(120)를 생성한다(S2000).
- [0061] 즉, 상기 에러 이미지(120)는 이미지 압축에 의해 상기 원본 이미지(100)가 변화된 정도이다.
- [0062] 예를 들어, 상기 원본 이미지(100)의 1행 1열의 픽셀 값이 '121'이고, 상기 압축 이미지(110)의 1행 1열의 픽셀 값이 '117'이라면 상기 에러 이미지(120)의 1행 1열 픽셀 값은 '4'가 된다.
- [0063] 다음, 상기 에러 이미지(120)로부터 베이스 매트릭스(130, Base Matrix)를 생성한다(S3000).
- [0064] 또한, 상기 베이스 매트릭스(130)는 상기 에러이미지(120)와 동일한 크기로 제작되며, 특정 비트 수를 원소로 한다.
- [0065] 또한, 상기 베이스 매트릭스(130)의 각 비트 수는 비트 조합의 경우의 수가 상기 에러 이미지(120)의 해당 픽셀 값을 넘지 않는 최대의 비트 수이다.
- [0066] 예를 들어, 상기 에러 이미지(120)의 픽셀 값이 '5'인 경우, 두 개의 비트는 네 가지 경우의 수(00,01,10,11)의 비트 조합을 갖고, 세 개의 비트는 여덟 가지 경우의 수의 비트 조합(000,001,010,011,111,110,101,100)을 가지므로 상기 베이스 매트릭스(130)의 원소의 값은 '2'가 된다.
- [0067] 또한, 상기 베이스 매트릭스(130)는 서로 중첩되지 않는 복수 개의 베이스 매트릭스 블록들이 상기 에러 이미지(120)와 동일한 크기의 하나의 베이스 매트릭스를 구성할 수 있다.
- [0068] 즉, 상기 베이스 매트릭스(130)는 상기 에러 이미지(120)의 각 화소 위치와 대응하는 위치에 원소를 갖는 상기 베이스 매트릭스 블록들의 원소는 동일한 행렬의 번호를 갖는다.
- [0069] 또한, 상기 베이스 매트릭스(130)은 아래의 수학적 식 1에 의해 계산된다.

수학적 식 1

$$Base_{i,j} = \lfloor \log_2(|ErrorImg_{i,j}|+1) \rfloor + OEF$$

- [0070]
- [0071] 여기서, $Base_{i,j}$ 는 상기 베이스 매트릭스(130)의 i행, j열의 비트 수, $ErrorImg_{i,j}$ 는 상기 에러 이미지의 i행, j열의 픽셀 값, OEF(optimal extension field)는 최적 확장체 값이다.
- [0072] 예를 들어, 상기 에러 이미지(120)의 1행 1열의 값이 '4'일 경우, 상기 베이스 매트릭스(130)의 1행 1열의 값은 '4'의 절댓값에 '1'을 더하고 ' $\log_2$ '를 취하면, '2.321928094887362'가 계산되고, 계산 값의 바닥함수(floor function), 즉, 최소정수값은 '2'가 된다.
- [0073] 또한, 상기 최적 확장체 값은 원하는 임베딩 비율(embedding rate)에 따라 변화되는데 임베딩 비율이 1[bpp]까지는 '0'이고, 2[bpp] 이상이 되면 증가한다. 본 발명에서는 상기 최적 확장체 값이 '0'인 경우를 상정하여 도시하였다.
- [0074] 다음, 임베딩하고자하는 비밀정보를 상기 원본 이미지(110)에 임베딩하여(S4000), 비밀 정보가 임베딩된 스테고 이미지를 생성한다(S5000).
- [0075] 또한, 상기 비밀정보는 상기 베이스 매트릭스(130)의 비트 수에 기반하여 임베딩된다.
- [0076] 다시 말해서, 상기 베이스 매트릭스(130)는 상기 비밀정보를 임베딩하기 위한 임베딩 위치와 임베딩량을 결정하는 역할을 한다.
- [0077] 또한, 본 발명에서는 상기 베이스 매트릭스(130)의 비트 수가 '2'이상인 행렬의 위치와 대응하는 상기 원본이미지

지(100)의 위치에 비밀정보를 임베딩한다.

- [0078] 또한, 상기 비밀정보를 임베딩하는 과정(S4000)은 먼저, 상기 비밀 정보를 특정한 길이의 비트 블록으로 분할한다.
- [0079] 도 2는 상기 비밀정보(200)를 보여주는 것으로 상기 비밀정보(200)는 n비트의 정보로 이루어진다.
- [0080] 또한, 상기 비밀정보(200)는 일정한 길이의 비트 블록(210)들로 분할되며, 상기 비트 블록(210)들은 서로 중첩되지 않는다.
- [0081] 또한, 분할되는 크기가 작을수록 신호대 잡음 비(PSNR)는 감소하고, 클수록 신호대 잡음 비(PSNR)는 증가한다.
- [0082] 다시 말해서 분할되는 크기가 클수록 노이즈는 작아진다.
- [0083] 본 발명에서는 상기 비트 블록(210)이 32bit의 길이를 갖는 비트 블록이 되도록 분할하였다.
- [0084] 다음, 상기 비밀정보(200)의 각 비트 블록(210)을 임베딩 값으로 변환한다(S4200).
- [0085] 또한, 상기 임베딩 값은 상기 원본 이미지(100)를 스테고 이미지로 변환하기 위한 정보이며, 상기 임베딩 값 자체가 상기 원본 이미지(100)에 더해지거나 빼지는 값은 아니다.
- [0086] 또한, 상기 임베딩 값을 생성하기 위해서는 먼저, 상기 비트 블록(210)을 십진수(300)로 변환한다(S4210).
- [0087] 도 3에 도시한 바와 같이 상기 비트 블록(210)은 '0000000000000000000000001101001'의 값을 갖고 이는 십진수 '105'와 같다.
- [0088] 다음, 상기 십진 값(300)을 임베딩 값(400)으로 생성한다(S4220).
- [0089] 또한, 상기 십진 값(300) '105'는 임베딩 값 {0,3,0,4}로 생성된다.
- [0090] 도 4를 참조하여 상세히 설명하면, 먼저, 십진 값 '105'를 베이스 매트릭스(130a)의 임베딩 위치들(B1,B2,B3,B4) 중 최초 임베딩 위치(B1)의 비트 수 '3'으로 나누어 몫(d) '35'와 나머지(d') '0'을 계산한다(S4221).
- [0091] 여기서, 나머지(d') '0'은 상기 베이스 매트릭스(130a)의 B1의 위치와 대응하는 상기 원본 이미지(100)의 위치에 임베딩될 임베딩 값이다.
- [0092] 또한, 도 4에서는 상기 베이스 매트릭스(130a)가 8×8메트릭스인 것으로 상정하였으며, 비트 수가 '2'이상인 원소('3','4','2','5')는 네 개뿐인 것으로 설정하였다.
- [0093] 즉, 상기 베이스 매트릭스(130a)의 2행 5열, 2행 7열, 2행 8열, 3행 1열과 대응하는 상기 원본 이미지(100)의 위치에 비밀정보가 임베딩되는 것이다.
- [0094] 다음, 이전에 계산된 몫(d)인 '35'를 다음 임베딩 위치(B2)의 비트 수 '4'로 나누어 몫(d1) '8'과 나머지(d'1) '3'을 계산한다(S4222).
- [0095] 여기서 나머지(d'1) '3'은 상기 베이스 매트릭스(130a)의 B2의 위치와 대응하는 상기 원본 이미지(100)의 위치에 임베딩될 임베딩 값이다.
- [0096] 다음, 이전에 계산된 몫(d1) '8'을 다음 임베딩 위치(B3)의 비트 수 '2'로 나누어 몫(d2) '4'와 나머지(d'2) '0'을 계산한다(S4223).
- [0097] 여기서 나머지(d'2) '0'은 상기 베이스 매트릭스(130a)의 B3의 위치와 대응하는 상기 원본 이미지(100)의 위치에 임베딩될 임베딩 값이다.
- [0098] 다음, 이전에 계산된 몫(d2) '4'를 다음 임베딩 위치(B4)의 비트 수 '5'로 나누어 몫(d2) '0'과 나머지(d'4) '4'를 계산한다(S4224).
- [0099] 여기서 나머지 '4'는 상기 베이스 매트릭스(130a)의 B4의 위치와 대응하는 상기 원본 이미지(100)의 위치에 임베딩될 임베딩 값이다.
- [0100] 또한, 상기 임베딩 값들을 계산하는 과정은 피제수인 이전에 계산된 몫보다 제수인 비트 수가 최초로 커질 때까지 반복된다.
- [0101] 즉, 상기 베이스 매트릭스(130a)의 B4 위치 다음에 비트 수가 '2' 이상인 임베딩 위치가 존재하더라도 임베딩

값을 생성하는 과정은 종료된다.

[0102] 다시 말해서, 상기 십진값(300) '105'를 임베딩 하기 위해서는 상기 원본 이미지(100)의 8×8블록 내에서 네 개의 픽셀만이 이용될 수 있고, 임베딩 값(400)은 {0,3,0,4}이고, 임베딩 되는 위치는 상기 베이스 매트릭스(130a)의 B1,B2,B3,B4와 대응하는 상기 원본 이미지(100)의 위치이다.

[0103] 다음, 상기 원본 이미지(100)에 상기 임베딩 값(400)을 임베딩한다(S4300).

[0104] 도 5를 참조하면, 상기 임베딩 값(400)을 임베딩 하는 방법은 크게 상기 임베딩 값(400)이 '2' 이상인 경우와 '2' 미만인 경우로 구분되며, '2' 이상인 경우, 임베딩 값에 해당 임베딩 위치에 대응하는 원본 이미지의 픽셀 값을 베이스 매트릭스의 해당 비트 수로 나눈 나머지를 감한 제1 변환 값, 상기 제1 변환 값에 해당 비트 수를 더한 제2 변환 값 또는 상기 제1 변환 값에 해당 비트 수를 감한 제3 변환값을 원본 이미지의 픽셀 값에 더하여 스테고 픽셀로 변환함으로써 임베딩이 수행되며, 아래의 수학적 2의 함수로 계산할 수 있다.

수학적 2

$$D = d' - \text{mod}(Original_{pixel(i,j)}, Base_{bit(i,j)})$$

$$\text{if, } - \lfloor \frac{Base_{bit(i,j)} - 1}{2} \rfloor \leq D \leq \lfloor \frac{Base_{bit(i,j)} - 1}{2} \rfloor \text{ then, } ChangeValue = D$$

$$\text{Else if, } 1 - Base_{bit(i,j)} \leq D \leq - \lfloor \frac{Base_{bit(i,j)} - 1}{2} \rfloor \text{ then, } ChangeValue = D + Base_{bit(i,j)}$$

$$\text{Else if, } \lfloor \frac{Base_{bit(i,j)} - 1}{2} \rfloor < D < Base_{bit(i,j)} \text{ then, } ChangeValue = D - Base_{bit(i,j)}$$

$$Stego_{pixel(i,j)} = Original_{pixel(i,j)} + ChangeValue$$

$$\text{if, } Stego_{pixel(i,j)} < 0 \text{ then, } Stego_{pixel(i,j)} = Stego_{pixel(i,j)} + Base_{bit(i,j)}$$

$$\text{Else if, } Stego_{pixel(i,j)} > 255 \text{ then, } Stego_{pixel(i,j)} = Stego_{pixel(i,j)} - Base_{bit(i,j)}$$

[0105]

[0106] 여기서, d'은 임베딩 값, $Original_{pixel(i,j)}$ 는 해당 원본 이미지 픽셀, $Base_{bit(i,j)}$ 는 해당 베이스 매트릭스의 비트 수, $\text{mod}(a,b)$ 는 a를 b로 나눈 나머지를 계산하는 함수, ChangeValue는 해당 원본 이미지 픽셀에 더해져 스테고 픽셀을 생성하기 위한 변환 값이다.

[0107] 즉, 상기 제1 변환 값은 "D"이고, 상기 제2 변환 값은 "D+Base_{bit(i,j)}"이고, 상기 제3 변환 값은 "D-Base_{bit(i,j)}"이다.

[0108] 또한, 상기 수학적 2에서도 알 수 있듯이, 각각의 조건에 따라 상기 제1 변환 값, 상기 제2 변환 값 및 상기 제3 변환 값 중 어느 하나의 변환 값이 해당 원본 이미지 픽셀에 더해질 변환 값으로 결정된다.

[0109] 예를 들어, 도 5의 예시 표(500)에서 원본 이미지(100a)의 픽셀 값이 '123'인 경우 임베딩 값은 '4'이고, 베이스 매트릭스의 픽셀 값은 '5'이다. 이 경우, D는 "4-mod(123,5)"로 계산되어 '1'이 된다.

[0110] 또한, D의 값인 '1'는 첫 번째 if의 경우와 같이 '-2'보다는 크고 '2'보다는 작은 값이므로 변환 값은 '1', 즉, 제1 변환 값이 더해져 스테고 픽셀의 값은 '124'가 된다.

[0111] 이러한 기법은 임베딩 값들을 원본 이미지에 픽셀에 각각 임베딩한다는 의미로 모듈식(modular) 함수 기법으로 정의하며, 썬(Thien) 및 린(Lin)[19]이 제시한 기법을 참조할 수 있다.

[0112] 또한, 스테고 픽셀 값 '214'를 임베딩 값 '4'로 디코딩하기 위해서는 스테고 픽셀 값 '214'를 간단히 베이스 매트릭스의 픽셀 값 '5'로 나누어 계산할 수 있고, 계산된 나머지 값이 임베딩 값 '4'가 된다.

[0113] 즉, 스테고 이미지를 수신한 수신인이 원본 이미지를 보유하고, 발신인으로부터 스테고 이미지와 이미지 품질 계수 및 최적 확장체 값을 전송받는다면 스테고 이미지로부터 임베딩 값을 구하고 임베딩 값들로부터 비밀 정보를 디코딩할 수 있다.

[0114] 더욱 자세하게는 수신인은 전송받은 이미지 품질 계수와 최적 확장체 값으로 원본 이미지를 압축하여 여러 이미

지를 생성하고, 에러 이미지로부터 베이스 매트릭스를 계산할 수 있으므로 스테고 이미지의 픽셀 값을 베이스 매트릭스의 비트 수로 나누어 나머지 값으로부터 비밀 정보를 얻어낼 수 있는 것이다.

- [0115] 또한, 상기 임베딩 값(400)이 '2' 이상인 경우 상기 수학적 식 2에 따른 기법을 사용하지 않고, 아래의 수학적 식 3과 같이 해당 임베딩 위치의 원본 이미지 픽셀 값에 상기 임베딩 값을 빼고, 베이스 매트릭스의 해당 비트 수로 나누어, 나눈 몫보다 크거나 같은 정수 중 가장 작은 정수에 상기 베이스 매트릭스의 해당 비트수로 곱하고, 상기 임베딩 값을 더하여 스테고 픽셀로 변환함으로써 임베딩 값을 원본 이미지에 임베딩할 수 있다.

수학적 식 3

$$Stego_{pixel(i,j)} = \left\lfloor \frac{Original_{pixel(i,j)} - d'}{Base_{bit(i,j)}} \right\rfloor \times Base_{bit(i,j)} + d'$$

- [0116]
- [0117] 예를 들어, 도 5에서 원본 이미지 값(100a)이 '123'인 경우, 스테고 픽셀의 값은 원본 이미지의 픽셀 값 '123'에 임베딩 값 '4'를 빼고, 비트 수 '5'로 나눈 최소정수값 '23'을 비트 수 '5'로 곱하고, 임베딩 값 '4'를 더한 '119'가 된다.

- [0118] 이 경우에도 수신인은 스테고 픽셀 '119'에서 비트 수 '5'를 나눈 나머지 '4'를 임베딩 값으로 디코딩하고, 임베딩 값들을 비밀정보로 변환할 수 있다.

- [0119] 이 방법은 변환 방정식 이용 기법으로 정의하며, 모듈식 함수 기법과 비교하여 연산량이 적은 장점이 있다.

- [0120] 그러나, 상기 임베딩 값이 '2' 미만인 경우, 최하위 비트 매칭 기법(LSB matching)을 이용하여 원본 이미지의 해당 픽셀값을 스테고 픽셀로 변환한다.

- [0121] 또한, 상기 최하위 비트 매칭 기법은 값이 '2'미만인 한 쌍의 임베딩 값이 존재할 때 수행되며, 먼저, 임베딩 값이 '2'미만인 임베딩 위치의 한 쌍의 원본 이미지 픽셀 값을 읽어들이는.

- [0122] 예를 들면, 도 5에서 임베딩 값이 '0'인 원본 이미지(100a)의 픽셀 값 '87'(이하 제1 픽셀 값이라 함), '165'(이하 제2 픽셀 값이라 함)를 읽어들이는.

- [0123] 다음, 두 개의 임베딩 값 중 제1 임베딩 값 '0'(원본 이미지의 픽셀 값 '87'에 임베딩 될 값)이 상기 제1 픽셀 값의 최하위 비트 값과 동일할 경우, 만약, 상기 제1 픽셀 값을 '2'로 나누고, 나눈 값의 바닥 함수 값에 상기 제2 픽셀 값을 더한 값의 최하위 비트가 제2 임베딩 값(원본 이미지의 픽셀 값 '165'에 임베딩 될 값)과 동일하지 않다면, 상기 제2 픽셀은 상기 제2 픽셀의 픽셀 값에 '1'을 더하거나 뺀 값인 스테고 픽셀로 변환되고, 그 외의 경우에는 상기 제2 픽셀은 상기 제2 픽셀과 동일한 값과 동일한 값인 스테고 픽셀로 변환되며, 상기 제1 픽셀은 상기 제1 픽셀과 동일한 값인 스테고 픽셀로 변환된다.

- [0124] 그러나, 도 5의 예시에서처럼 제1 임베딩 값 '0'이 상기 제1 픽셀 값 '87'의 최하위 비트 값(십진수 '87'을 2진수로 변환하였을 때 최하위 비트 값은 '1'이다)과 서로 상이할 경우, 이 경우에는 만약, 상기 제1 픽셀 값에 '1'을 뺀 값을 '2'로 나누고, 나눈 값의 바닥함수 값에 상기 제2 픽셀 값을 더한 값의 최하위 비트 값이 상기 제2 임베딩 값과 동일하다면, 상기 제1 픽셀은 상기 제1 픽셀의 픽셀 값에 '1'을 뺀 값의 스테고 픽셀로 변환되고, 그 외의 경우 상기 제1 픽셀은 상기 제2 픽셀과 동일한 값의 스테고 픽셀로 변환되며, 상기 제2 픽셀은 상기 제2 픽셀과 동일한 값의 스테고 픽셀로 변환된다.

- [0125] 도 5의 예시로 예를 들면, 상기 제1 임베딩 값은 상기 제1 픽셀 값의 최하위 비트 값과 서로 상이하고, 상기 제1 픽셀 값 '87'에 '1'을 뺀 '86'을 '2'로 나눈 바닥 함수값 '43'에 상기 제2 픽셀 값 '165'를 더한 값의 최하위 비트 값은 '0'이므로 상기 제2 임베딩 값과 동일하다.

- [0126] 즉, 상기 제1 픽셀 값 '87'은 '87'에서 '1'을 뺀 '86'의 값을 갖는 제1 스테고 이미지로 변환되고, 상기 제2 픽셀 값은 상기 제2 픽셀 값 '165'와 동일한 값을 갖는 제2 스테고 이미지로 변환된다.

- [0127] 또한, 수신인이 상기 제1 스테고 픽셀과 상기 제2 스테고 픽셀을 디코딩할 때는 상기 제1 스테고 픽셀의 값 '86'의 최하위 비트인 '0'을 상기 제1 임베딩 값으로 디코딩할 수 있고, 상기 제2 스테고 픽셀의 값 '165'는 상기 제1 스테고 이미지 값 '86'을 '2'로 나눈 바닥 함수 값 '43'에 상기 제2 스테고 픽셀의 값 '165'를 더한

'208'의 최하위 비트인 '0'을 상기 제2 임베딩 값으로 디코딩할 수 있다.

[0128] 아래의 표 1은 본 발명의 스테고 이미지 생성방법에 의해 생성된 스테고 이미지와 종래의 에지 적응적(EA:Edge adaptive) 이미지 스테가노그래피 기법[5]과 HUGO기법[6]으로 생성된 스테고 이미지의 비밀정보 검출 가능성에 대한 실험결과를 보여주는 표이다.

[0129]이 검출 가능성에 대한 실험은 2차 감산 픽셀 인접성 모델(SPAM:subtractive pixel adjacency model)[7]을 통해 이루어졌고, 2차 감산 픽셀 인접성 모델은 비밀정보 검출 가능성이 낮을수록 보다 큰 평균 검사 에러를 출력한다.

[0130]또한, 원본이미지로써 BOSS 버전 1.01의 이미지 데이터베이스를 이용하였으며, 인텔 코어(Intel Core) i5-2500, 3.3-3.6GHz, 8GB RAM의 컴퓨터 장치와 함께 MATLAB R2012a 통해 실험을 수행하였다.

표 1

원본이미지(10,000개의 이미지)		평균 검사 에러		
임베딩 비율 (Payload)(bpp)	임베딩 용량 (Capacity)(bit)	EA기법[5]	HUGO기법[6]	본 발명(PSNR, IQF, OEF)
0.05	13,101	0.4717	0.5000	0.4782(65.19dB, 96.0, 0)
0.1	26,214	0.4309	0.4844	0.4504(62.29dB, 94.0, 0)
0.2	52,428	0.3381	0.4469	0.4066(59.46dB, 90.0, 0)
0.3	78,643	0.2549	0.4010	0.3506(57.83dB, 85.0, 0)
0.4	104,857	0.1920	0.3600	0.2477(56.63dB, 75.0, 0)

[0132]표 1에서도 알 수 있듯이 본 발명의 일 실시예 따른 스테고 이미지의 평균 검사 에러가 EA기법[5]보다 높아 검출가능성이 더 낮은 것을 알 수 있으며 HUGO기법[6] 만큼의 검출 불가능성을 갖는 것을 알 수 있다. 그러나, EA 기법[5], HUGO기법[6]이 최대 1[bpp]까지 임베딩 할 수 있는 반면, 본 발명의 스테고 이미지 생성방법은 최대 4[bpp]까지 비밀정보를 임베딩할 수 있는 장점이 있다.

[0133]또한, HUGO기법[6]의 경우 0.4[bpp]로 비밀정보를 임베딩하기 위해서 약 147.05초가 걸렸으나 본 발명의 스테고 이미지 생성 방법은 단 3.5초가 걸려 실행속도 측면에서 매우 우수한 것을 알 수 있었다.

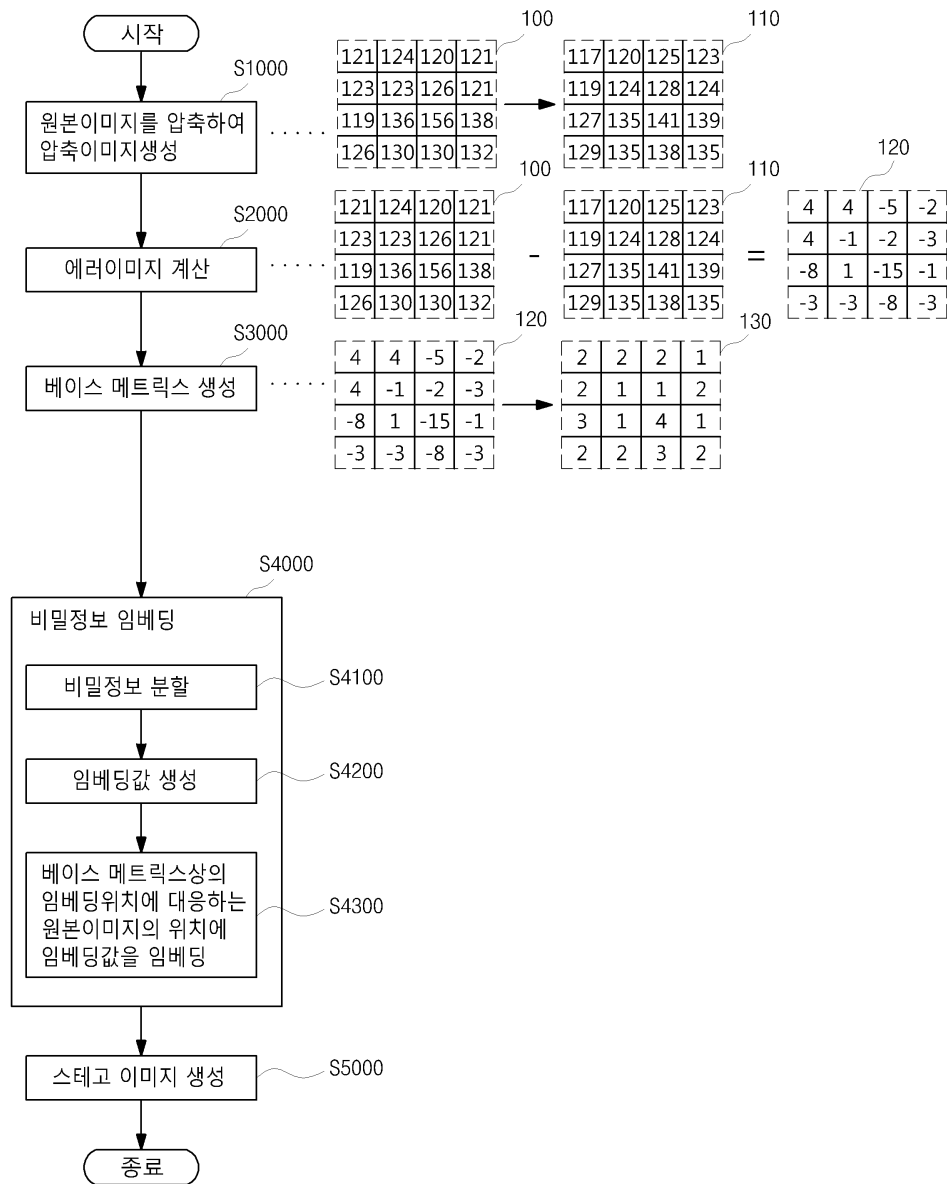
[0134]아래의 표 2는 본 발명과 종래의 기법들을 이용하여 BOSS 버전 1.01의 원본 이미지 중, 개코 원숭이(Baboon), 레나(Lena), 호수(Lake), 남자(man), 후추(Peppers) 이미지에 비밀정보를 임베딩한 후, 스테고 이미지의 신호대 잡음비(PSNR(dB))를 평가한 표이다.

표 2

임베딩 비율 (bpp)	임베딩 용량 (bits)	원본 이미지	임베딩기법	PSNR (dB)
1.68	440,000	Baboon	수정된 PVD[10]	40.07
			본 발명(IQF=2, OEF=0)	49.03
1.68	440,000	Lena	BPCS 기법[12]	34.60
			PVD 기법[9]	41.70
			본 발명(IQF=1, OEF=0)	48.79
2.52	660,725	Baboon	Side-Match 기법[11]	33.53
			본 발명(IQF=0.3, OEF=0)	44.38
2.5	655,360	Lake	제키 등의 기법[13]	27.52
			과라 등의 기법[14]	34.59
			본 발명(IQF=0.2, OEF=0)	44.71
2.82	740,000	Man	MBNS 기법[15]	38.10
			본 발명(IQF=0.01, OEF=4)	40.81

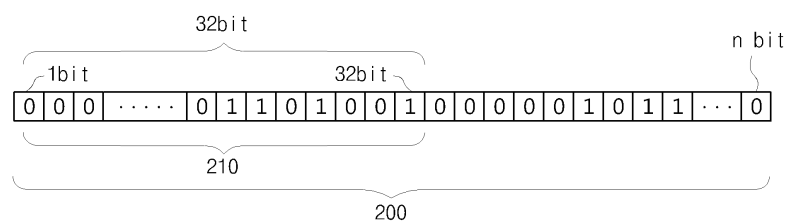
도면

도면1

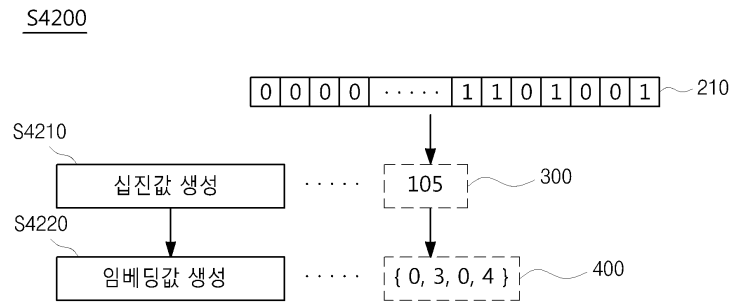


도면2

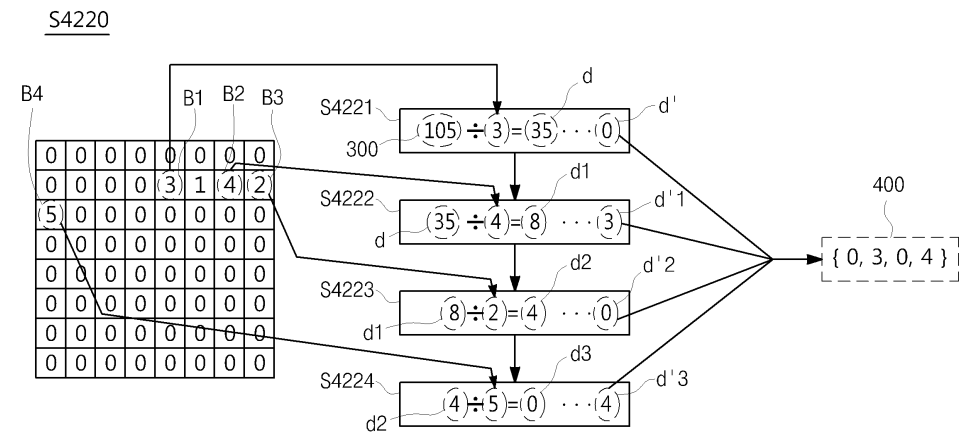
S4100



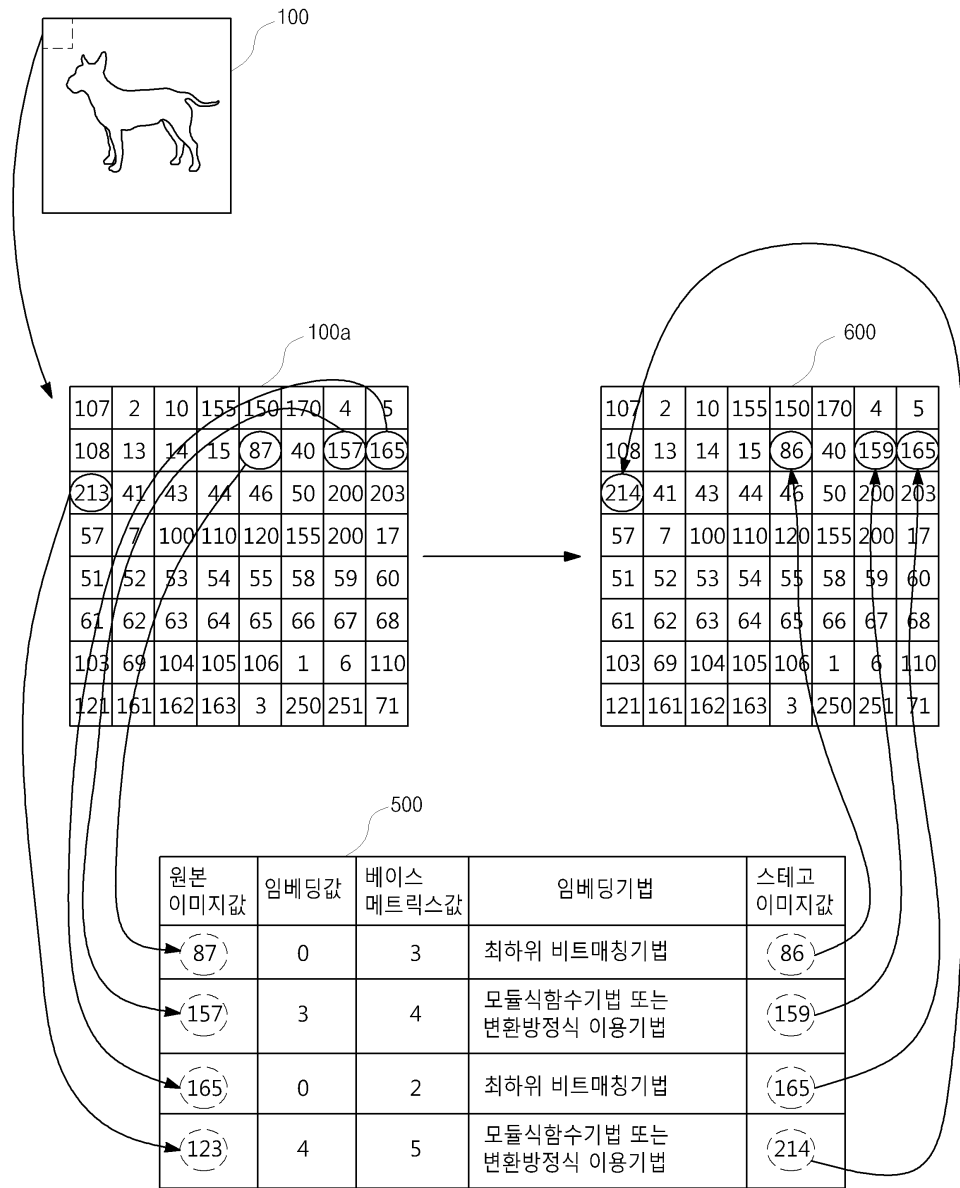
도면3



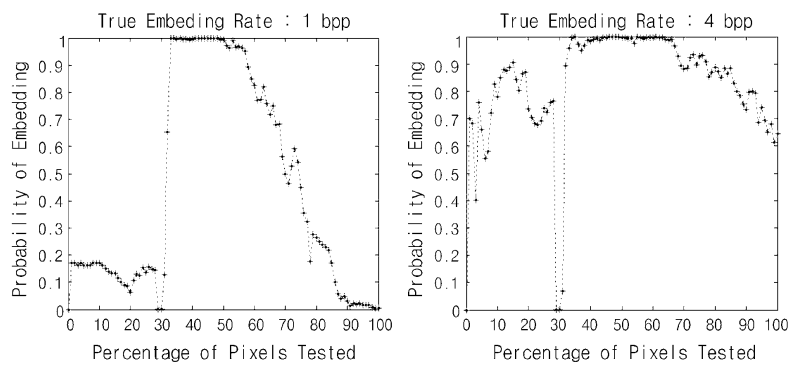
도면4



도면5



도면6



도면7

