확장된 파트너 교환을 이용한 복원 가능한 정보 숨김 기술

*우재현 **김형중 ***바실리 ****최수정

고려대학교

*bull0330@korea.ac.kr

Advanced Changing Partners Techinque in Reversible Steganography

*Woo, Jae-Hyeon **Kim, Hyoung-Joong ***Sachnev, Vasiliy ****Choi, Su-Jeong

Korea University

요약

기존의 Steganography는 정보를 숨김에 있어 히스토그램을 shifting 하거나 픽셀간의 상관관계를 변형하는 등 다양한 방식이 제안되었다. 그런 논문들이 가지고 있던 용량적 한계를 일부 극복하고, PSNR을 고려하여 새로운 방식을 제안한다. 소개하는 기술은 두 색깔을 파트너로 맺어, 숨김 비트에 따라 서로 파트너로 바뀌도록 설계되었다. 숨긴 용량의 크기와 PSNR의 변화를 실험 결과에서 볼 수 있으며, 이는 기존 논문과의 비교를 통해 효율성이 입증된다.

1. 서론

스테가노그래피 방식은 원본 이미지에 특정 메시지를 숨기고자 하는 방식이다. 과거의 스테가노그래피 방식에서는 숨기는 메시지만 중요하게 다루어졌기 때문에 LSB에 값을 일괄적으로 숨기려는 메시지로 바꾸는 등의 방식을 사용하였으나, 현재는 원본 이미지의 복원도 중요시 되고 있는데, 이를테면 원본 이미지의 무결성이나 전자서명을 원본 이미지에다 숨기는 경우이다. 그림을 전송할 때, 원래 이미지에 워터마킹을 삽입하고서 합쳐진 그림을 전송한다. 그러면 수신측에서는 원래 그림과 워터마킹을 모두 복원하여, 이 정보들로부터 무결성과 인증을 입증할 수가 있다.

최근 들어 복원 가능한 스테가노그래피나 워터마킹 기술이 많이 연구되어 졌다. Chang은 side matching and relocation 방식을 제안했는데, 이는 한 비트를 VQ-compressed 인덱스에 숨긴다. 예를 들어, 4×4 픽셀로 이루어진 한 블록을 가정한다면, 한 블록에 비트를 숨겨 블록정보가 바뀌더라도, 왼쪽 블록과 위쪽 블록과의 경계면에서의 픽셀들의 차이를 이용하여 원래 그 블록의 인덱스를 찾아볼 수 있다는 사실을 이용하여 비트를 숨긴다. Ni는 shifting histogram 방식을 제안했는데, 이는 원본 이미지의 히스토그램의 일부를 한 칸씩 옮겨 빈 색깔을 만든 후, 이곳을 이용하여 비밀정보를 숨기는 방식이다. 전혀 사용되지 않는 색이 있으면 이를 zero point라 명하고, 히스토그램에서 최대값인 색깔을 peak point라 한다. Peak point와 인접한 색깔에서부터 zero point 까지 색깔들을 하나씩 옮기게 되면 peak point 바로 인접한 색이 새로운 zero point가 된다. 이 방식은 peak point 에 비트를 숨기는데, 비트값이 0이면 그대로 두고, 비트값이 1이면 peak point를 바로 인접한 zero point로 색을 바꾼다. 이외에도 로케이션 맵을 이용하는 다양한 방식들이 연구되고 있다. Tian은 두 픽셀의 값 차이를 이용하여, 이를 두 배로 만든 뒤 여기에 비트를 더하는 방식을 제안했다. 0과 255를 넘는 overflow를 방지하기 위해 일정한 조건에 따라 값 차이를 EZ, EN, CN, NC 등으로 분류하고, EZ, EN, CN 그룹에 비트를 숨긴다. EN과 CN은 비트를 숨긴 후 겹치는 부분이 생기기 때문에 복원할 때 이를 구분해서 복원하기 위해 로케이션 맵을 사용한다. 두 픽셀의 값 차이를 두 배로 하게 되면 항상 그 값이 짝수가 되는데, 여기에 비트 0을 숨기면 그대로 짝수가 되나, 비트 1을 숨기면 홀수로 바뀌게 된다.

이 논문에서는, 기존의 방식과는 전혀 다른 방식을 제안하여 복원에 필요한 정보를 가급적 줄이는 방식을 제안함으로써, 숨길 수 있는 용량을 늘일 수 있는 새로운 방식을 제안한다.

2. 파트너 교환

가. 기본 알고리즘

자연계에 존재하는 화면을 담은 이미지라면, 픽셀들과 그 인접 픽셀을 비교해 볼 때 차이가 크지 않다. 그림 1을 보자.

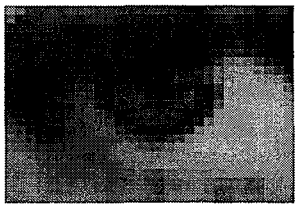
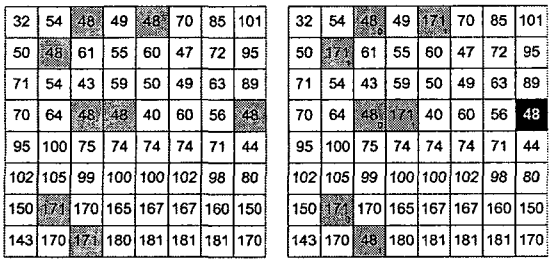


그림 1. 확대된 Lena 영상의 일부

검은 눈동자와 흰색 눈자위가 만나더라도 그 경계면에서는 검은색에서 흰색으로 급격히 변화하지 않고 중간의 점증적 변화과정이 있다. 이러한 사실을 일반화 시키면, 한 픽셀에 대해 그 인접 픽셀은 랜덤하다기 보다는 특정 상한이나 하한을 갖게 된다.

이제 그림 2에서의 예제를 통해 알고리즘의 원리를 설명하기로 한다.



(a) 원본 이미지 (b) 숨긴 이미지

그림 2. 픽셀의 예

그림 2의 (a)는 숨기기 이전의 원본 이미지이다. 각 값들은 gray scale에서 가지는 0~255까지의 색을 의미한다. 이 그림에서 픽셀 색이 48인 경우와 171인 경우를 보자. 48값을 가진 픽셀의 오른쪽 픽셀만 고려해 본다면, 그 값은 49, 70, 61, 48, 40 이 되고, 마지막 48값의 픽셀은 오른쪽 픽셀을 가지고 있지 않으므로 고려하지 않는다. 마찬가지로 171값을 가진 픽셀의 오른쪽 픽셀은 170, 180이 된다. 48의 오른쪽 픽셀들 중 최대값은 70이 되고, 최소값은 40이며, 171의 오른쪽 픽셀들 중 최대값은 170이고 최소값은 180이다. 여기서, 이러한 오른쪽 픽셀들의 최대, 최소값을 Maxright와 Minright를 사용하여 다음과 같이 표기하기로 한다.

$$\text{Maxright}(48) = 70, \quad \text{Minright}(48) = 40$$
$$\text{Maxright}(171) = 180, \quad \text{Minright}(171) = 170$$

여기서, Minright(171) > Maxright(48) 에 주목하자. 48과 171은 오른쪽 픽셀의 범위가 서로 겹치는 부분이 없다. 우리는 이렇게 범위가 서로 겹치지 않는 두 색을 하나의 쌍으로 만들 수 있고, 서로를 파트너라고 칭한다. 이제, 일반적인 두 색 i, j를 다음과 같이 한 쌍의 파트너로 만든다.

$$\text{쌍의 집합} = \{(i, j) \mid i < j,\ \text{Maxright}(i) < \text{Minright}(j)\}$$

여기서, 각 i, $j(i < j)$에 대해 Maxright(i) 값을 경계라고 하고, i, j를 멤버색이라고 한다. i값의 픽셀과 그 오른쪽 픽셀은 모두 경계 이하의 값을 가지며, j값의 픽셀과 그 오른쪽 픽셀은 모두 경계보다 값이 크다. 즉 48값의 픽셀과 그 오른쪽 픽셀은 모두 70 이하이며, 171값의 픽셀과 그 오른쪽 픽셀은 모두 70보다 크다는 것을 알 수 있다. 우선 (48, 171)의 한 쌍만 가진 경우를 고려하기로 하자. 비트열을 숨기기 위해, 첫 픽셀부터 오른쪽 방향으로 검색을 시작한다. 픽셀값이 48이거나 171이면, 비트값을 숨기게 되는데, 비트값이 0이면 픽셀값을 그대로 두고, 비트값이 1이면 픽셀값을 그 파트너값으로 변환시킨다. 비트열이 '0110101'일 경우의 결과가 그림 2의 (b)에 나와 있다. 검게 표시된 48값의 픽셀은 오른쪽 픽셀이 없으므로 아무 정보도 숨겨지지 않았다.

복원할 때는, 반대 방향으로 진행한다. 즉 마지막으로 비트를 숨긴 픽셀부터 첫 픽셀까지 거꾸로 비트를 찾아내고 원본이미지를 복구한다. 픽셀값이 48일 경우, 그 오른쪽 픽셀이 70보다 크다면 숨김 비트는 1이 되고 48을 그 파트너인 171로 복구시키고, 그 오른쪽 픽셀이 70 이하라면 숨김 비트는 0이 되고 48은 그대로 둔다. 픽셀값이 171일 경우, 그 오른쪽 픽셀이 70보다 크다면 숨김 비트는 0이 되고 171은 그대로 두며, 그 오른쪽 픽셀이 70 이하라면 숨김 비트는 1이 되고 171은 그 파트너인 48로 복구시킨다. 지금까지는 오른쪽 픽셀만 예로 들었으나, 아랫방향, 오른쪽 아래 대각선 방향도 모두 가능하며, 용량을 늘이기 위해 이를 혼합하여 사용할 수도 있다. 이렇게 비트를 원본 이미지에 숨기고 다시 복원시키기 위해서는 오직 (i, j, 방향, 경계값)의 값들만 필요하며, 한 방향만 사용하게 되면 방향 정보도 필요가 없다.

때로는 원본 이미지에서 사용되지 않는 색깔을 이용하여 쌍을 만들 수 있다. (i, j)에서 i 혹은 j가 원본이미지에서 사용되지 않은 색일 수 있는데, 이 경우는 경계값이 필요가 없다. 예를 들어 i가 사용되지 않은 색일 경우, i는 원본 이미지에 있던 어떤 색깔과도 쌍을 만들 수 있으며, 비트를 숨길 때 j 값을 만나면, 비트가 0일 경우 그대로 두고 비트가 1일 경우 i로 바꾼다. 복원 과정에서는, 픽셀값이 j이면 비트값 0을 찾아내고 픽셀값은 그대로 두며, 픽셀값이 i이면 비트값 1을 찾아내고 픽셀값은 파트너인 j로 복원시킨다. 이 경우에는 (i, j, 방향)의 값이 필요하며, 역시 한 방향만 사용할 경우 방향 정보도 필요없다.

이렇게 쌍의 집합을 만드는 방법은 두 가지 모드가 있다. 첫 번째 모드는 *최대 용량* 모드로서, 원본이미지에서 가장 많이 사용되었던 색들을 먼저 쌍으로 만드는 것이다. 이것은 가장 많이 사용되었던 색이, 그 다음 많이 사용된 색 중에서 쌍을 만들 수 있는 색을 찾아 쌍의 집합에 포함시킨다. 한 쌍이 찾아지면, 쌍의 집합에 포함되지 않았던 색 중에서 다시 반복적으로 가장 많이 사용되었던 색부터 자신의 파트너를 찾는다. 이 모드는 숨기고자 하는 비트량이 많을 때 사용한다.

두 번째는 *PSNR* 모드로서, 가장 많이 사용되었던 색부터 자신의 파트너를 찾긴 하나, 자신의 파트너를 그다음으로 많이 사용된 색부터 찾는 것이 아니고 자신과 비슷한 인접색깔들부터 찾게 된다. 마찬가지로 한 쌍이 찾아지면, 쌍의 집합에 포함되지 않았던 색 중에서 가장 많이 사용되었던 색부터 다시 같은 방식으로 자신의 파트너를 찾아 쌍의 집합에 포함시킨다. 이 모드는 숨기고자 하는 비트량이 작을 때 이미지의 왜곡을 최소화 시키기 위한 모드이다.

어느 모드이건, 비트를 숨기고 복원시키는 방식은 동일하다. 숨기는 과정에서는 처음부터 멤버색을 찾으며, 비트에 따라 비트가 0이면 그대로 두고 비트가 1이면 그 파트너로 변경한다. 복원 과정에서는 비트가 숨은 마지막 픽셀부터 거꾸로 멤버색을 검색하되, 멤버색과 그 오른쪽(또는 헤더에 포함된 방향으로) 색깔이 경계를 기준으로 같은 방향에 있는지 다른 방향에 있는지 판단하여, 같은 방향이면 비트 0을 찾고 픽셀은 그대로 두며, 다른 방향이면 비트 1을 찾고 픽셀은 그 파트너로 다시 복원시켜 준다.

나. 확장 알고리즘

지금까지는 두 개의 색깔만으로 쌍의 집합을 만드는 방식을 설명했다. 두 개의 색깔로 된 쌍에는 한 픽셀에 한 비트만 숨길 수 있으나, 최대 용량 모드의 변형으로서 그 용량을 늘이기 위해 2개 이상의 비트를 숨기려면 쌍의 형태를 다음과 같이 바꿀 수 있다.

$$\text{확장된 쌍의 집합} = \{(i, j, k, l) \mid i \text{ 는 } j, k, l \text{ 과 모두 겹침이 없음}\}$$

i가 j와 겹침이 없다는 것은 $i < j$ 일 경우 Maxright(i) < Minright(j)가 되고, $i > j$ 일 경우 Maxright(j) < Minright(i)가 성립한다는 뜻이다. 이 경우는 경계값이 두 가지가 나올 수 있는데, 경계선은 다음 각각의 경우를 나누어 설정한다.

1) i가 (i, j, k, l) 중에서 최소값일 경우

2) i가 (i, j, k, l) 중에서 최대값일 경우

3) i가 (i, j, k, l) 중에서 최대, 최소값이 아닐 경우

1)의 경우 경계 = Maxright(i)

2)의 경우 경계 = Max(Maxright(j), Maxright(k), Maxright(l))

3)의 경우 두 개의 경계값이 필요하며 다음과 같이 설정한다.

예를 들어, $j < i < k < l$ 일 경우, j와의 경계는 Maxright(j)이 되고, k, l과의 경계는 Maxright(i)가 된다. $j < k < i < l$ 이라고 가정을 하면, j, k와의 경계는 Max(Maxright(j), Maxright(k))이 되고 l과의 경계는 Maxright(i)가 된다.

이렇게 i가 겹치지 않는 3가지 색과 쌍을 맺게 되면, i는 2비트씩 숨길 수 있으며, 각 2비트 값이 00이면 i 그대로, 01이면 j로, 10이면 k로, 11이면 l로 바뀌게 된다. 숨김 과정에서는 i 색만 찾아 비트를 숨기게 되고 (j, k, l)에는 따로 숨김 과정이 없다. 복원 과정에서는 (i, j, k, l)를 모두 찾아서, 픽셀값이 i가 아닐 경우 경계값과의 비교를 통해 비트가 숨겨졌을 경우 이를 복원하고 픽셀값은 i로 복원시킨다. 확장 알고리즘에서는 일방적으로 i에만 비트를 숨기게 되므로, j는 다시 확장된 쌍 (j, a, b, c)를 만들 수 있고, j에 같은 방식으로 2 비트씩 숨길 수 있다. k, l도 마찬가지로 쌍을 만들 수 있다.

3. 결 과

가. 기본 알고리즘

실험은 그림 3에 나타난 6 가지 그림을 사용하였다.

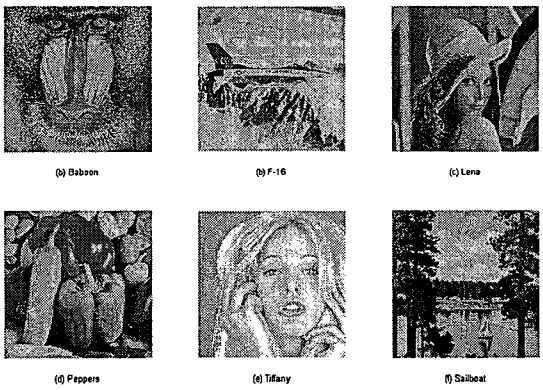


그림 3. 샘플 이미지

기본 알고리즘에서는 최대 용량 모드와 PSNR 모드가 있음을 이미 언급한 바 있다. 최대 용량 모드는 가장 많이 쓰이는 색깔 순으로 쌍을 만들었기 때문에 PSNR은 그다지 좋지 않다. 그림 4에서 보이는 결과는 최대 용량 모드에서의 결과인데, 최대 용량의 90% 이상을 숨기게 되면 대체적으로 10dB 전후의 성능을 보인다. 하지만 그 용량은 두 가지 경우를 제외하고 0.73 bpp(bit per pixel) 이상이 된다.

하지만 크기가 작은 메시지라면 최대 용량 모드를 사용할 필요가 없다. 최대한 거리가 가까운 색깔을 쌍으로 맺어줌으로써 왜곡 현상을 최소화 하도록 쌍을 맺어주는 PSNR 모드를 사용할 수 있다. 그림 5에서는 PSNR 모드의 결과를 보여주고 있다. 두 모드 모두 복원 정보인 헤더의 크기는 숨김 메시지에 따라 50 ~ 300 바이트 가량 된다.

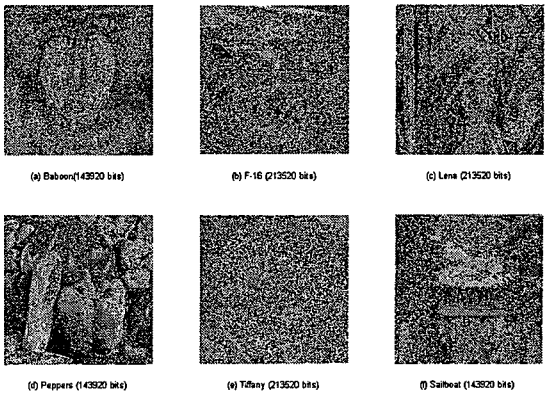


그림 4. 최대 용량 모드에서 용량의 90% 이상 숨긴 결과

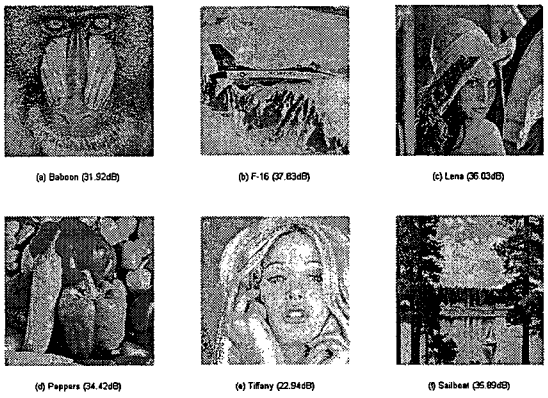


그림 5. PSNR 모드에서 10928 bit를 숨긴 결과

나. 확장 알고리즘

확장 알고리즘에서는 PSNR 모드를 고려하지 않는데, 앞서 설명한 바와 같이 그 용량을 더 늘이기 위한 방식이기 때문이다. 하지만 항상 최대 용량 모드보다 더 용량이 커지는 것은 아니다. 확장 알고리즘에서의 결과는 그림 4와 차이가 나지 않으므로 그림으로의 비교는 생략하기로 하고, 표 1에서 bpp 비교를 통해 용량 변화에 따른 결과를 비교해 보면, F-16과 Tiffany의 경우는 1.0 bpp를 넘는 것을 알 수 있다. 즉 이미지 그림보다도 더 많은 용량의 메시지를 숨길 수가 있다.

표 1. 최대 용량 모드에서의 최대 bpp

이미지	최대 용량 모드의 최대 bpp	확장 알고리즘의 최대 bpp
Baboon	0.555	0.406
F-16	0.866	1.136
Lena	0.875	0.776
Pepper	0.595	0.513
Tiffany	0.923	1.007
Sailboat	0.732	0.697

4. 결 론

이 논문에서는 기존의 방식들과는 차이가 나는 새로운 숨김 기술을 제안하였다. 서로 다른 색깔을 한 쌍으로 맺어, 비트에 따라 서로의 파트너로 색깔을 바꿔주는 방식이다. 이 방식의 가장 큰 특징은 기존에 제시된 방식들에 비해 숨길 수 있는 메세지의 용량이 늘었음을 알 수 있는데, 그 값은 평균적으로 0.8 bpp보다 크다. 이는 두 가지로 설명할 수 있다. 첫 째, 원본 이미지에서 많이 사용되었던 색깔이 쌍을 이룰 경우 그 색깔이 사용된 픽셀만큼의 비트를 숨길 수 있다는 것이고, 두 번째로는 헤더의 크기가 로케이션 맵에 비해 현저히 줄어들어 로케이션 맵이 차지했던 불필요한 공간을 활용할 수 있다는 것이다. 하지만 용량이 커질수록 이미지의 왜곡 현상이 심하므로, 이를 보완할 수 있도록 쌍을 만드는 방식을 추가로 변경시켜야 할 것이다.

5. 참고 문헌

1. Woo, J. J., Kim, H. J., Sachnev, V., Choi, S. J. : Changing Partners Technique in Reversible Steganography. IJWISA, (2007) 23-31
2. Chang, C. C., Lin, C. Y. : Reversible Steganography for VQ-Compressed Images Using Side Matching and Relocation. IEEE Transactions on Information Forensics and Security, vol. 1, no. 4, (2006) 493-501
3. Ni, Z., Shi, Y. Q., Ansari, N., Su, W. : Reversible Data Hiding. IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 3. (2006) 354-362
4. Xuan, G., Yao, Q.,, Yang, C., Gao, J., Chai, P., Shi, Y. Q., Ni, Z., : Lossless Data hiding Using Histogram Shifting Method Based on Integer Wavelets. Lecture Notes in Computer Science, vol. 4283 (2006) 323-332
5. Tian, J. : Reversible Data Embedding Using a Difference Expansion. IEEE Transactions on Circuits and Systems for Video Technology (2003) 890-896
6. Kamstra, L., Jeijmans, H. : Reversible Data Embedding Into Images Using Wavelet Techniques and Sorting IEEE, vol. 14 (2005) 2082-2090
7. Fridrich, J., Goljan, M., Du, R. : Invertible authentication Proceedings of the SPIE Photonics West, Security and Watermarking of Multimedia Contents III, San Jose, California, vol. 3971, (2001) 197-208
8. Fridrich, J., Goljan, M., Lisonek, P., Soukal, D. : Writing on wet paper, IEEE Transactions on Signal Processing, vol. 53 (2005) 3923-3935
9. Alattar, A. M. : Reversible watermark using difference expansion of triples, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1 (2003) 501-504
10. Alattar, A. M. : Reversible watermark using difference expansion of quads, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3 (2004) 377-380
11. Alattar, A. M. : Reversible watermark using difference expansion of generalized integer transform I, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 13, no. 8 (2004) 1147-1156