

은닉형 광고 디스플레이 제품 정보 표현 기법

최영환*, 황인준*

*고려대학교 전자전기공학과

e-mail : {work48, ehwang04}@korea.ac.kr

Steganography-based Product Information Representation Scheme for Digital Signage

Young-Hwan Choi*, Eunjung Hwang*,

*School of Electrical Engineering, Korea University

요 약

최근 들어, 정보 광고 디스플레이(Digital Signage)는 영상 재현 및 IT 기술 발전으로 비약적 성장을 거듭하고 있다. 기업들의 마케팅이나 광고, 트레이닝 효과 및 고객 경험을 유도할 수 있는 커뮤니케이션 도구로써 공항이나 호텔, 병원 등 공공 장소에서 방송 프로그램뿐 아니라 특정한 정보를 함께 제공하는 디지털 영상 장치로써 다양하게 사용되고 있다. 현재는 기존 상업용 디지털 정보 디스플레이(DID)에 주요 기능을 제어할 수 있는 소프트웨어나 관리 플랫폼까지 종합적으로 공급하는 형태로 시장에 공급되고 있다. 현재 거의 대부분의 프로그램은 플래시를 이용하여 정보를 일방적으로 전달하는 모습을 하고 있어 사용자의 요구나 다양한 정보의 표현에 제약이 많았다. 본 논문에서는 사용자와의 쌍방향 통신을 유도를 통하여 광고의 예술성뿐 아니라 정보제공의 기능까지도 동시에 만족시킬 수 있는 은닉형 광고 디스플레이 제품 정보 표현 기법을 제안한다. 구체적으로 예로써 Histogram shifting 을 이용한 QR 코드 은닉을 통해 영상에서 사용자가 원하는 제품에 대한 QR 코드를 숨기고 표출하는 방법을 보인다.

1. 서론

최근 들어 광고 시장의 경쟁이 치열해지면서 정보 광고 디스플레이는 영상 재현 및 IT 기술 발전과 함께 비약적 성장을 거듭하고 있다. 정보 광고 디스플레이의 진화 방향은 크게 Technology 와 Creative 라는 양대 축을 기준으로 진화 중에 있다. Digital Signage 는 광고의 역할에 대한 인식의 전환으로 일반 사용자에게 보다 생활에 밀접하게 접근하여 광고를 할 수 있다는 장점을 가지고 있다. 또한 영상 디스플레이의 비약적인 성장으로 저렴한 가격으로 높은 광고 효과를 올릴 수 있게 되었다.

하지만 지금까지의 정보 광고 디스플레이는 일방적으로 정보를 보여 주는 역할에 충실하여 사용자의 요구나 다양한 정보의 표현에 제약이 많았다. 따라서 본 논문에서는 단순히 보여주는 광고를 넘어서 쌍방향 통신을 유도함으로써 사용자의 호기심을 자극하여, 더 큰 광고 효과를 유도하고자 한다. 제안하는 기술은 영상에 다양한 형태로 표시되어 있는 객체에 대해 그 종류 및 특성을 정확하게 파악하여 해당 객체에 대한 부가 정보를 자동으로 기존 이미지에 삽입하는 기술이다. 구체적인 기술 적용 예로써 사용자에게 간단하며 많은 정보를 줄 수 있는 웹 기반의 QR 코드를 부가 정보로 이미지에 삽입하는 기법을 상세히 기술한다. QR 코드를 삽입하는 방법으로 영상 정보 은닉(Image Steganography) 기술을 이용하여, 객체 이미지

자체에 QR 코드의 정보를 숨기는 방식 개발을 목표로 한다.

논문의 주요 부분은 다음과 같이 이루어진다. 2 장에서는 관련 연구를 설명하고, 3 장에서는 실제로 사용된 정보 은닉 기법에 대하여 설명을 한다. 4 장에서는 3 장에서 제안한 방법으로 정보 은닉을 하였을 때의 실험 결과를 보여 주고 5 장에서 실험 결과에 따른 결론을 도출한다.

2. 관련 연구

정보 은닉 기술에는 다양한 방법이 존재 한다. 기존의 연구로는 Least Significant Bit (LSB)[1,2,3]와 같은 영상에서 각 픽셀의 회색 조 영상 값을 8bit 로 표현한 후, 4bit 의 Least significant bits 를 숨기고자 하는 bit stream 의 4bit 으로 교체하는 방법이다. 이 방법은 영상 왜곡이 심해질 수 있다는 단점이 있다.

다른 정보 은닉 방법으로 Histogram Shifting[4,5,6,7]을 들 수 있다. 이 방법은 영상의 Histogram 에서 피크 값과 피크 값 이후에 존재하는 영의 값을 찾아서, 피크 값 이후부터 영 사이에 존재 하는 모든 히스토그램을 이동하게 된다.

Histogram Shifting 후, 순차적으로 이미지를 스캔 하며 피크 값을 이미지에서 찾고, 숨기려 하는 bit stream 에서 1 을 만날 경우 피크 값에 1 을 더하게 된다. 예를 들어, 피크 값이 154 이면, 알고리즘은 이미

지를 스캔하며 154 를 찾게 된다. 만약, 숨기고자 하는 bit stream 이 1 이면, 이미지를 스캔 하며 찾은 154 의 픽셀 값을 155 로 증가를 시켜 주게 된다.

알고리즘 1: Histogram Shifting

```

Input: Image I(m,n), grayscale value of  $G \in [0, \dots, 255]$ ,
          histogram of image H(x)
          hiding stream bits B(t)
Output: integer value peakVal,
           integer value minVal


---

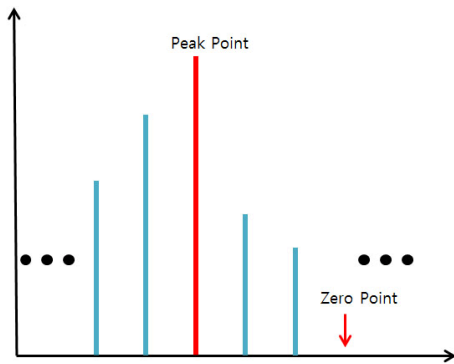

1: peakVal  $\leftarrow 0$ ;
2: for each point in H(xi) where  $0 \leq i \leq 255$ 
3:   if H(xi) is peak point
4:     peakVal  $\leftarrow$  H(xi);
5:   end if
6:   if H(xi) is minimum point and H(xi) > peakVal
7:     minVal  $\leftarrow$  H(xi);
8:   end if
9: end for
10: for each point H(xi) where peakVal+1  $\leq i \leq$  minVal-1
11:   H(xi+1)  $\leftarrow$  H(xi)
12: end for
13: for each bits in B(tj) where  $0 \leq j \leq E$ 
14:   if B(tj) is equal to 0
15:     continue
16:   else if B(tj) is equal to 1
17:     for each pixel value G in I(m,n)
18:       if G is equal to peakVal
19:         G  $\leftarrow$  peakVal + 1
20:       end if
21:     end for
22:   end if
23: end for
E denotes end of hiding stream bits
    
```

알고리즘 1 은 bit stream 을 숨기는 과정을 보여 주고 있다. 우선 알고리즘은 히스토그램 내에 존재 하는 피크 값과 영의 값을 히스토그램 전역을 스캔 하며 찾게 된다. 그 후 피크 값 +1, 영의 값 +1 사이에 있는 모든 값에 1 을 더해 shifting 을 한다. 그 후 bit stream 을 scan 하여 1 을 만나게 되면 이미지 상에서 피크 값을 가지는 픽셀을 순차적으로 찾아 1 씩 값을 더 하게 된다. 이 과정은 bit stream 의 끝까지 계속 반복 되게 된다.

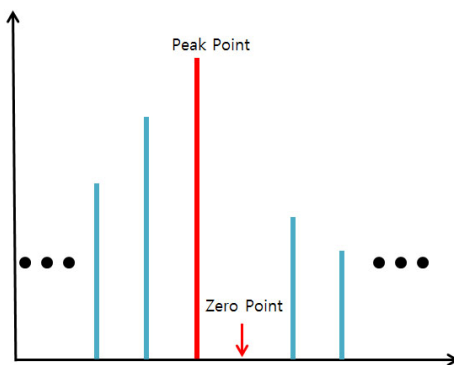
숨겨 놓은 bit stream 을 다시 정보 추출을 할 때에는 이미지를 스캔하며, 피크 값을 만났을 경우 0 을 추출 하고 피크 값 + 1 의 값을 만났을 경우 1 을 추출 하여 숨긴 bit stream 을 다시 뽑아 내게 된다. 이미지를 복원 하는 방법은 이미지 전역을 스캔 하여 피크 값 + 1 의 픽셀 값을 만나게 되면, 다시 -1 을 해주어 정상 값으로 돌려주고, 또한 <피크 값 + 2 , 영의 값> 사이에 존재 하는 값에도 -1 을 해주면 영상의 다시 복원 된다.

3. 다구간 Histogram Shifting

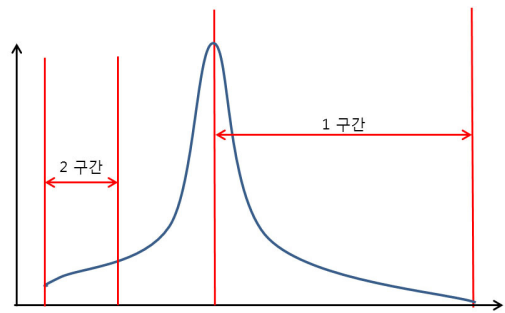
기존의 Histogram Shifting 방법의 문제는 피크 값이 가지는 개수만큼 bit 을 저장 할 수 있었다. 숨기려는 이미지가 대용량인 경우 저장할 수 있는 한계가 존재하였다. 제안하는 방법은 다중 Point 를 이용한 Histogram Shifting 방법으로써, 사람의 시선을 최소화한 으로 방해하는 Histogram 의 최저 값들을 합쳐 한번 더 Shifting 을 수행하여 더 많은 값을 저장하는 방법 이다.



(가) Histogram Shifting 전



(나) Histogram Shifting 후
(그림 1) Histogram Shifting



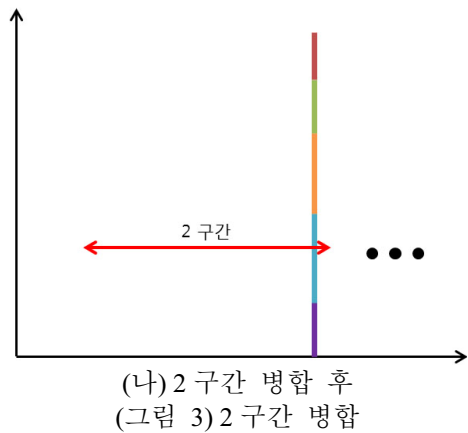
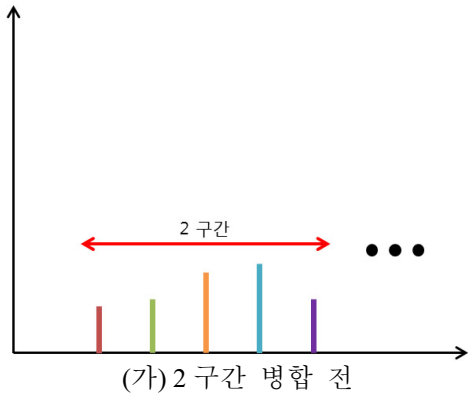
(그림 2) 다 구간 Histogram Shifting

그림 2 와 같이 1 구간에서는 기존의 Histogram Shifting 방법을 적용하여, bit stream 의 일정 부분을 숨기게 되고, 1 구간에서 처리 안 되는 나머지 bit stream 부분을 2 구간에 숨기게 하는 것이다.

하지만 2 구간에서의 피크 값은 큰 값을 가지지 않으므로, 숨길 수 있는 양이 한정되어 있다. 그래서 사람이 변화가 일어나도 알아볼 수 없을 정도에 구간을 지정하여, 그림 2 와 같은 상황에서는 2 구간에서 가장 큰 픽셀 값이 되는 곳에 나머지 픽셀 값들의 모든 개수를 더해 주게 된다.

그림 3(나)와 같이 2 구간을 병합 시킨 후, 기존의 방법을 적용하여, 1 구간에서 남은 bit stream 을 숨기게 된다. 숨겨진 Bit 를 추출할 때도, 기존의 방법과 비슷하게 2 구간에서 피크 값을 만날 때는 0 을 피크 값-1

의 값을 만나게 되면 1 을 추출 하여, bit stream 을 완성 하게 된다.



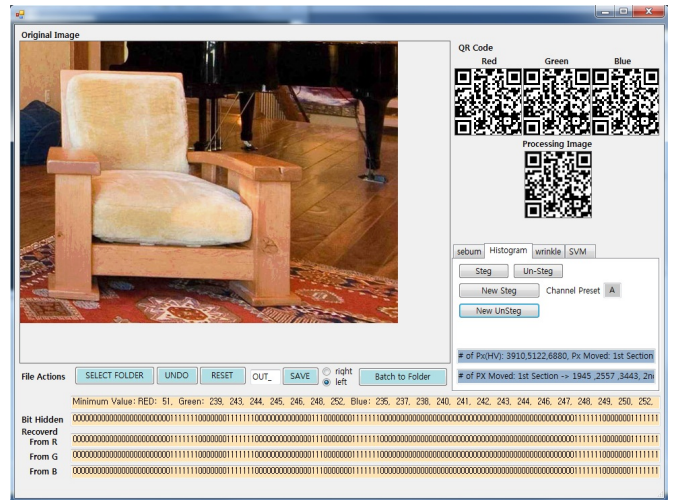
4. 실험

그림 4 는 실험을 위한 프레임워크를 보여주고 있다. 개발에 사용한 언어는 C#, .net framework 이며, 영상 처리 라이브러리는 openCV 를 C# 환경에 이식한 emguCV[8]를 사용하였다.

실험에서는 RGB 의 세 개의 각 채널에 대하여 동

적으로 구간을 정하게 진행하였다. 동적 구간 설정의 조건은 피크 값 구간에 존재하지 않고, 1 구간의 정보는 엔닉 방법에 영향을 끼치지 않는 곳으로 자동 설정되게 하였으며, 0~255 인 색상 레벨, 즉 256 개의 색상 정보를 각 16 개로 나눠 총 16 구간으로 나누어 진행하였다. 엔닉 된 정보를 다시 추출 하기 위한 키 값이 저장 되어야 한다. 저장 되는 키 값은 <피크 값, 영의 값, 구간 값>으로 저장이 되며, 이 내용은 BMP 의 영상 헤더에 저장이 되어 추출 할 때 사용 된다.

그림 4 의 우측 상단 3 개의 이미지는 각 채널에서 복원된 QR 코드를 보여주고 있으며, 우측 하단에 존재하는 QR 코드는 기존의 이미지를 보여주고 있다. 사용된 QR 코드는 100 x 100 이며, 이미지는 0 과 255 두 가지의 색상으로 이루어져 있으므로, bit stream 으로 변환하여 이미지에 삽입하였다.



(그림 4) 프레임워크

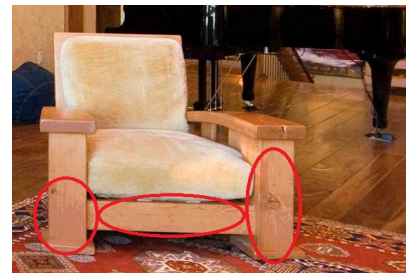
그림 5 에서 제품을 표현할 수 있는 일부분의 이미지를 잘라내어 실험을 진행하였다. 전체 이미지는 HD TV 의 해상도 사이즈인 1920 x 1080 이미지에서



(그림 5) 예제 광고 이미지



(그림 6) Red 채널 정보 엔닉



(그림 7) Green 채널 정보 엔닉



(그림 8) Blue 채널 정보 엔닉



(그림 9) 모든 채널 정보 엔닉



(그림 10) 정보 추출 후 이미지

그림 5 정도의 제품 이미지를 잘라 내어 사용 하였다. 그림 5의 크기는 561 x 419이다.

그림 6은 Red 채널에 정보를 은닉 후의 이미지이다. 그림 5와 6을 비교하면 약간 밝아진 톤이 보이긴 하나 미세한 것을 확인할 수 있다. 그림 7은 Green 채널에 정보를 은닉한 후의 이미지를 보여주고 있다. 빨간 원으로 표시된 부분이 가장 큰 이미지 왜곡을 보여주고 있는 부분이다.

그림 8은 Blue 채널에서 정보 은닉 후의 이미지이다. Red 채널과 마찬가지로 기존의 이미지와 큰 차이를 보여주지 않으며, 오히려 Red 채널보다 더욱 완전하게 삽입된 것을 볼 수 있다. 그림 9는 모든 채널에 정보 삽입 후의 이미지이다. Green 채널의 영상 왜곡에 의해 모든 채널을 사용하였을 때는 많은 데이터가 저장 가능하나 왜곡이 존재함을 보여주고 있다.

그림 10은 정보 추출 후의 이미지이다. 오른쪽 하단에 추출된 QR 코드를 확인할 수 있다. 복원된 후의 이미지 또한 기존의 이미지와 큰 차이가 없음을 보여주고 있다. 그림 10의 이미지 삽입과 복원은 Blue 채널을 통해 진행을 하였다.

5. 결론

본 논문에서는 광고 이미지에 QR 코드를 삽입한 후, 사용자의 요구에 의해 QR 코드를 이미지로부터 추출하여 보여 주는 방법을 제안하고, 이미지 정보 은닉 방법에 의하여 진행된 결과가 시각적인 이미지 왜곡 없이 진행됨을 입증하였다. 실험 결과 Green 채널에서 보이는 영상 왜곡을 제외하고, 시각적인 왜곡이 보이지 않았다. 여러 가지의 제품 이미지를 통해 실험을 진행한 결과, 각 이미지마다 왜곡을 일으키는 채널이 존재하는걸 확인하였다. 이번 실험 결과를 토대로 자동으로 왜곡이 최소화 되는 채널을 선택하여 정보를 삽입하고 추출하는 시스템을 Digital Signage 분야에 개발하여 적용할 계획이다.

Acknowledgement

“본 연구는 지식경제부 및 정보통신산업진흥원의 IT/SW 창의연구과정의 연구결과로 수행되었음” (NIPA-2011-C1820-1102-0018) "본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음" (NIPA-2012-H0301-12-3006)

참고문헌

- [1] N.F. Johnson, S. Jajodia, Exploring steganography: seeing the unseen, Computer Practices (1998) 26–34.
- [2] J. Fridrich, M. Goljan, and R. Du, "Invertible Authentication," Proc. SPIE Security and Watermarking of Multimedia Contents, (2001) 197-208.
- [3] C.H. Yang, C.Y. Weng, S.J. Wang, H.M. Sun, Adaptive data hiding in edge areas of images with spatial LSB domain systems, IEEE Trans. Inf. Forens. Security 3 (3) (2008) 488–497.
- [4] C.L. Tsai, H.F. Chiang, K.C. Fan, C.D. Chung, "Reversible data hiding and lossless reconstruction of binary images using pair-wise logical computation mechanism", Pattern Recognition 38 (11) (2005) 1993–2006.
- [5] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, "Reversible data hiding," IEEE Transactions on Circuits and Systems for Video Technology 16 (3) (2006) 354–361
- [6] P. Tsai, Y.C. Hu, H.L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," Signal Processing 89 (2009) 1129–1143
- [7] J. Wang, J. Ni, "A fast performance estimation scheme for histogram shifting based multi-layer embedding," Proc. Of IEEE 17th Int. Conf. on Image Processing, (2010) 26–29
- [8] EmguCV : cross platform .Net wrapper to the OpenCV, <http://www.emgu.com/wiki>