

8개의 점을 이용한 개선된 4중 임베딩

A New Reversible Watermarking Method Based on 4 times-Embedding Using 8 points

이 병 용*, 김 형 중*, Vasilii Sachnev*
 Byeong-yong Lee, Hyoung-Joong Kim, Vasilii Sachnev

Abstract

정보가 가역적으로 숨기어 졌다는 말은, 숨기고자 하는 정보가 삽입된 콘텐츠에서 비밀 정보를 찾아내고 또한 원본 콘텐츠를 완벽하게 복구할 수 있다는 말이다. 여러 가지 콘텐츠에 정보를 숨길 수 있으나 그 중에서 특히 이미지에 정보를 숨기고자 할 때는 두 가지 목표가 있다. 첫째는, 숨기고자 하는 데이터의 용량을 극대화 시키는 것이고 둘째는, 정보를 숨긴 다음에 생긴 이미지와 원본 이미지와의 차이를 줄이는 것이다. 본 논문은 데이터를 삽입시키는 차분값으로서 예측 오류 값을 이용하여 정보를 숨기는 개선된 방법을 제안하려고 한다.

Keywords : Reversible Data Hiding, Sorting, Prediction, Double Embedding.

I. 서 론

정보가 가역적으로 은닉되었다는 뜻은, 여러 가지 매체에 정보를 삽입한 뒤에 그 숨겼던 데이터와 원본의 콘텐츠를 손상 없이 복구할 수 있다는 뜻이다. 이런 기술들은 숨겨진 데이터뿐만 아니라, 원본의 이미지의 보존이 대단히 중요한 이미지들에게 사용되게 된다. 원본의 보존이 중요한 이미지로서 대표적인 것은 군사용 이미지와 의료 이미지가 있다.

정보를 가역적으로 숨기는 방법은 이미지 뿐만아니라 오디오와 다른 매체에도 적용될 수 있으나 본문에서는 이미지에 국한 시켜서 다루려고 한다.

앞서 말한 것과 같이 가역적으로 정보가 은닉시킨다는 것은 두 가지를 만족시켜야 한다. 첫째는 데이터의 완전한 복구이고, 둘째는 원본 콘텐츠의 완벽한 복원이다. 이 두 가지를 기본 요건을 만족시키면서 다음과 두 가지를 충족시킬 때 효과적으로 은닉되었다고 한다. 효과적인 은닉을 위한 두 가지 조건을 다음과 같다. 첫째는, 삽입 용량이 커야 한다. 둘째는, 은닉되는 데이터가 커지더라도 데이터가 삽입된 데이터와 원본 데이터의 왜곡이 크지 않아야 한다.

이 논문에서는 그동안 발표된 논문에 대해서 정리하고 개선점을 제시하고자 한다

2장에서는 은닉 기술이 어떻게 진보되어 왔는지를 정리하고 3장에서는 새로운 알고리즘을 제안하도록 하겠다.

II. 발표된 가역 임베딩 알고리즘

1. 두 개의 픽셀을 이용한 차분 확장

Tian은 [2] 가역 정보 은닉을 위해서 인접한 두 개의 픽셀을 하나의 단위로 보고 이들의 차를 이용해서 정보를 은닉한다. 각각의 픽셀의 밝기 값이 x 와 y 라고 하자. (1)식을 통해서 두 픽셀의 밝기값의 차분값(h)과 평균값(a)를 구할 수 있다. 그림 1에서와 같이 두 개의 픽셀의 단위를 $I_{i,j}$ 로 표시할 수 있다. 구한 값을 (3)식과 같이 2배로 확장시키고, 비트를 삽입하게 된다. 그리고 이 삽입된 비트값을 이용하게 해서 원본의 x, y 대신에 정보가 삽입된 X, Y 를 구할 수 있게 된다.

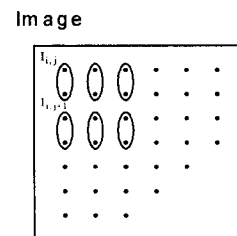


그림 1. 두 개의 픽셀의 페어(pair)

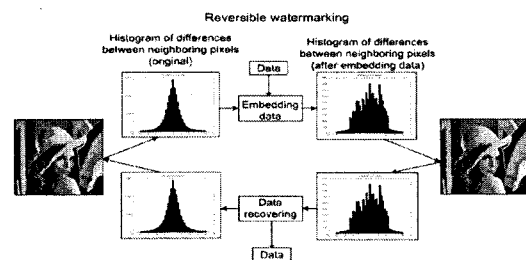


그림 2. 임베딩 후의 이미지 왜곡

접수일자 : 2009년 7월 21일
 최종완료 : 2009년 8월 14일
 *고려대학교 정보경영전문대학원 대학원
 **고려대학교 정보경영전문대학원
 교신저자, E-mail : khj-@korea.ac.kr

이때 그림 1에서 보는 바와 같이 원래의 차분 히스토그램과 데이터를 은닉후의 차이가 나게 된다. 그 변화된 분량이 왜곡된 분량이다.

$$h = x - y \quad (1)$$

$$a = \left\lfloor \frac{x+y}{2} \right\rfloor \quad (2)$$

$$H = 2h + bit \quad (3)$$

$$X = a + \left\lfloor \frac{H+1}{2} \right\rfloor, Y = a - \left\lfloor \frac{H}{2} \right\rfloor \quad (4)$$

(3)식에서 두 개의 픽셀의 차분에 2를 곱하는 것은, 컴퓨터가 정보를 2진수로 처리하기 때문이다. 차분에 2를 곱하게 되면은 맨끝 자리가 하나가 더 생기게 된다. 2를 곱하게 되면은 비트 이동(bit shift)가 발생하기 때문이다. 이 비트 이동이 된 자리 LSB(Least Significant Bit)에 은닉하고자 하는 정보를 삽입하는 방식이 Tian이 제시한 방식이다.

가역 임베딩에서 정보를 은닉시킨 후에 다시 은닉한 정보를 얻어내는 과정은 다음과 같다. 정보를 다시 추출하는 과정을 일반적으로 디코딩 과정이라고 부른다.

디코딩 시에는 (5)식과 같이 H를 다시 2로 나누는 방법을 택한다. 정보를 은닉할 때에 두 개의 픽셀의 차분값을 2배로 하고 그 값에 은닉하고자 하는 페이로드를 삽입했기 때문이다. 그 H값을 2로 나눈 값을 floor함수에 넣게 되면은 은닉된 정보는 버리고 원본의 x,y의 차분값만 남게 된다. 은닉되는 페이로드는 0 또는 1 인데, 2로 나눈후 floor 값을 취하게 되면은 소수값이 되어서 버려워지기 때문이다.

또한 삽입된 실제 페이로드를 구하는 식은 (6)과 같다.

$$h = \left\lfloor \frac{H}{2} \right\rfloor \quad (5)$$

$$bit = H \bmod 2 \quad (6)$$

Tian 방법은 최초로 제시된 것으로 의의가 있다. 그러나 Tian 방법은 두 개의 픽셀을 한 단위로 취급하여 정보를 은닉하기 때문에, 이론적으로 최대한 0.5bpp까지 은닉할 수 있다는 단점이 있다. 두 픽셀당 한 개의 bit만이 은닉된다는 것이다. 또한 또 하나의 단점은 데이터를 은닉할 때, 또 다른 데이터를 은닉해야 한다는 것이다. 픽셀의 밝기 값은 0에서 255의 값을 취하는데, 정보를 은닉한 다음에 이 한계점보다 아래의 값을 취하거나 혹은 위의 값을 취하는 경우가 있다. 이런 경우를 분별하기 위해서 데이터 위치 지도가 필요하다. 이 위치 지도는 픽셀이 원래의 값인지 혹은, 정보가 은닉된 픽셀인지를 분별해 주는 역할을 한다. 따라서 이론적으로 이 데이터 위치 지도의 크기는 전체 픽셀의 절반에 해당한다. 이 위치 지도를 압축하지 않고 데이터를 은닉하려고 한다면은 한 bit의 데이터도 압축할 수가 없다. Tian은 이를 극복하기 위해서 데이터 위치 지도를 압축해서 은닉하는 방법을 택했다.

이 데이터 위치 지도(location map)을 압축하는 방법은

여러 가지가 있는데 그 중에서도 Tian은 JBIG2를 이용한 압축을 택했다.

그 외에도 더 훌륭한 압축방법도 있지만, 상업상의 이유로 인해서 실용화 되지는 못했다.

2. 차분한 뒤에 정렬하는 방식

Tian은 두 개의 픽셀을 취해서 그것을 한 개의 단위로 그 안에 데이터를 은닉하는 방식을 택했다. 그리고 데이터를 은닉하는 순서는 단위 순서대로 했다.

이런 방식을 택하게 되면은, 원본 이미지와 데이터가 은닉된 이미지와의 왜곡이 커지는 현상이 나타난다. 두 픽셀의 차분이 적은 두 개의 픽셀을 갖고 데이터를 은닉하는 것이 차분이 큰 두 개의 픽셀을 갖고 데이터를 은닉하는 것보다 더 작은 왜곡을 일으키게 된다.

왜냐하면

$$H = 2h + bit$$

h(차분)에 2를 곱하기 때문이다. 이 곱한 값 H을 갖고 원본의 x,y를 X,Y로 변화시키기 때문에, 이 차분이 적은 수록 원본의 x,y와 데이터가 은닉된 X,Y의 왜곡이 적어지게 되는 것이다.

따라서 데이터를 은닉할 때, 이 차분이 적은 것부터 은닉하는 것이 더 적은 이미지 왜곡을 가지고 온다는 아이디어를 갖고 접근하기 시작했다.

Kamstra와 Heijmans의 [3] 방식도 Tian과 같이 두 픽셀을 한 개의 단위로 사용하고 그 안에 한 비트씩 숨기는 방식이다. 이 기본 아이디어에 위에서 설명한 왜곡을 적게 일으키는 단위에 먼저 데이터를 은닉하기로 했다. 이를 실행하기 위해서 kamstra는 한 셀의 평균값(a)과 주변 셀들의 평균값들 사이의 상관계수를 구하였다.

이 상관계수는 데이터 차분이 적은 단위에 먼저 데이터를 은닉시키기 위해서 구한 값이었다. 이 상관계수의 값은 가역성을 만족시켜야 했다. 상관계수의 값은 데이터가 은닉된 다음에도 그 셀의 평균값은 변하지 않는다는 특성을 이용해서 가역성을 만족시켰다.

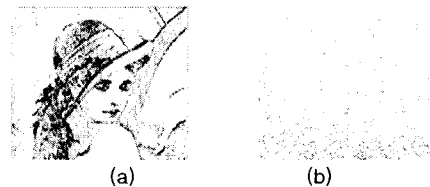


그림 3. (a) Tian 위치지도 (b) 정렬 후의 위치지도

또한 Tian은 데이터 위치 지도를 JBIG2를 통해서 압축시켰다. Kamstra도 동일한 방법을 사용했다. 이때에 데이터 위치지도의 값이 일정한 특성을 갖기 때문에 kamstra의 방법을 채택하게 되면은 JBIG2의 압축률이 높아지게 된다. 즉 더 적은 양의 데이터 위치 지도만은 은닉하면 된다는 것이다. 따라서 kamstra의 아이디어는 Tian의 아이디어에 비해서 두 가지 진보된 면이 있다.

첫째는, 데이터 위치 지도 자체의 용량이 줄어서 은닉할 데이터가 줄었다는 것이다.

둘째는, 차분 값이 적은 단위부터 데이터를 은닉하기 때

문에 데이터 왜곡이 적다는 것이다.

그러나 두 개의 픽셀을 한 단위로 취급해서 데이터를 은닉하기 때문에 이 방법은 결정적으로 0.5bpp를 넘을 수 없다는 한계가 있다.

3. 예측 오류값과 히스토그램 이동

앞의 방법들의 단점은 최대한 데이터를 은닉해도 0.5bpp를 넘을 수 없다는 것이다. 그 단점을 극복하기 위해서 예측값을 이용하는 아이디어를 내게 되었다. Thodi와 Rodriguez는 [4] JPEG-LS 예측 오류 기법을 사용한다. JPEG-LS는 데이터를 은닉하려는 목표 픽셀 밝기 값에 대한 예측값 구해서 그 차분을 이용하는 방식이다. 이렇게 구해진 픽셀값 x 에 대한 예측값이 \hat{x} 라고 했을 때, 예측 오류값(prediction error) d 는 (7)식과 같고, (8)식과 같이 데이터가 삽입된다.

$$d = x - \hat{x} \tag{7}$$

$$D = 2d + bit \tag{8}$$

앞선 방식들은 두 개의 픽셀을 한 단위로 사용하기 때문에, 결정적으로 0.5bpp를 넘을 수 없다는 한계가 있었다. 그러나 이 방식은 한 개의 목표 픽셀과 그 주변 픽셀의 예측값과의 차분을 이용하기 때문에 모든 픽셀에 데이터를 은닉할 수 있다는 장점이 있다.

그러나 이 방식도 한계가 있다. 앞서 Tian의 방식에서 데이터 위치 지도가 필요했던 것처럼, 이 방법도 반드시 위치 지도가 필요한 것이다. 이 위치 지도를 은닉하지 않고 순수하게 데이터만 은닉하기 위한 기술이 요청되었다. 그렇게 되면은 순수하게 데이터만 은닉해서 이론상으로는 1bpp까지 데이터를 숨길 수가 있다.

Thodi와 Rodriguez는 이 문제를 해결하기 위해서 새로운 기법을 도입했다. 히스토그램을 이동해서 해결하는 방식이다.

앞선 방법의 경우는 차분값이 0인 경우에 1 비트를 삽입하게 되면은 1이 될 경우, 원래 1인 값과 데이터를 은닉해서 1이 된 값을 구별하기 위해서 위치지도가 필요했다.

히스토그램 이동 기법은 다음과 같다. 경계값을 두 개 먼저 정한다. 이때 경계값은 하나는 양수로 하나는 음수로 정한다. 이 경계값안에 있는 값들은 모두 데이터가 은닉된 것으로 하기 위해서 양의 경계값에 1을 더해서 데이터를 이동시키고, 음의 경계값을 2배해서 데이터를 이동시킨다. 이런 과정을 거치면은 경계값안에 있는 차분값을 갖는 픽셀에만 데이터를 은닉하게 된다. 이 과정을 통해서 값이 겹치는 픽셀이 발생하지 않기 때문에, 겹치는 값을 구분하기 위해서 사용되었던 위치 지도가 필요하지 않게 된다.

예를 들어서 경계값이 1이라고 하면, 1이 아닌 값들은 차분값 2씩으로 이동시키는 과정을 통해서 데이터를 은닉한다. 그림 4가 이 과정을 보여 주고 있다.

물론, 밝기값 255와 0 사이의 값을 넘는 값에는 위치지도 값이 필요하지만 이런 값들을 갖는 픽셀은 거의 없기 때문에 위치 지도는 극히 일부만 필요하게 된다.

아래는 위에서 설명한 데이터 임베딩 과정과 그 은닉한 데이터를 추출해서 원본의 이미지를 회복하는 디코딩 과정을 기술한 것이다.

```

데이터 임베딩 과정
if |h| < T
H = 2h + Bit
else
H = sign(h)(|h|+T)
end
    
```

```

데이터 디코딩 과정
if |H| < 2T
h = floor ( H/2 ) ;Bit = H mod 2
else
h = sign(H)(H-T)
end
    
```

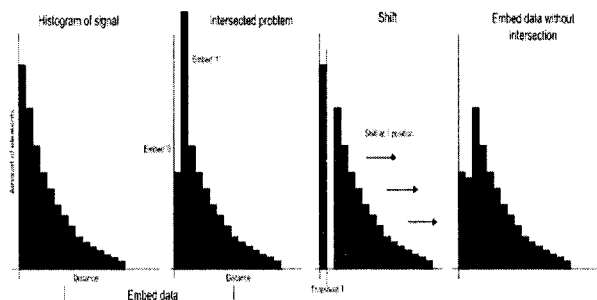


그림 4. 히스토그램의 이동

Thodi와 Rodriguez의 기법에서는 위치 지도를 은닉하지 않아도 되기 때문에 은닉하고자 하는 데이터의 최대의 삽입 용량이 이론상 1 bpp로 까지 상승했다.

그러나 이 방법도 한계가 있다. 최대 삽입 용량은 증가했고 위치지도도 줄었지만, 정렬을 사용할 수 없다는 것이다.

Kamstra는 정렬을 사용해서 데이터가 은닉된 후에 왜곡이 적은 점부터 데이터를 은닉했다. Thodi와 Rodriguez의 기법은 주위의 3개의 픽셀을 사용하기 때문에 그 값이 변형되면은 디코딩이 불가능해진다. 그렇기 때문에 정렬을 사용할 수가 없게 된다.

Thodi와 Rodriguez의 기법과 정렬의 기법을 동시에 사용하기 위해서는 원래의 순서대로 재정렬 시키는 방법을 찾아야 했다.

4. 예측 오류값 이용과 정렬의 응용

Vasiliy는 Thodi와 Rodriguez의 예측 오류 기법과 Kamstra의 정렬 기법을 사용하는 방법을 고안했다.

Thodi와 Rodriguez의 예측 오류 기법은 주변의 3개의 점을 이용하기 때문에 정렬을 통해서 순서가 어지럽히게 되면은 원래의 값을 복구할 수 없다는 단점이 있었다. 따라서 원래의 순서대로 재정렬시킬 수 있는 방법을 찾아야 했다. Vasily는 데이터를 두 개로 분리해서 이 문제를 해결했다. Vasily는 데이터를 X 세트와 ●으로 분리를 했다. 데이터를 은닉할 때는 X세트, ●세트, 디코딩 시에는 ●세트, X세트 순서로 실행한다.

이 때 X 세트는 "cross set", 그리고 • 세트는 "Dot set"이라고 부른다. 이미지 전체를 그림 5와 같이 cross set과 dot set으로 분류할 수 있다. 이 때 cross와 dot은 전혀 겹치지 않는다.

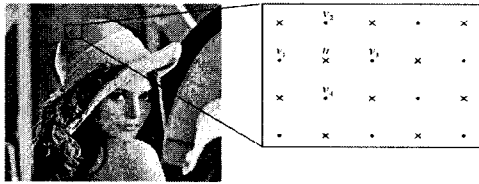


그림 5. 이미지 전체를 두 개의 세트로 나눔

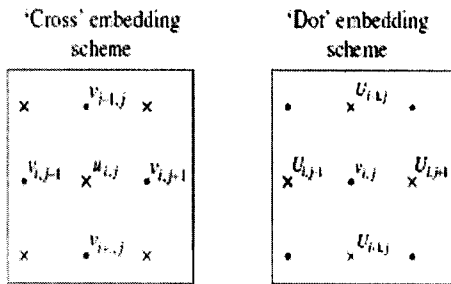


그림 6. X 세트와 • 세트

예측값은 (9)식과 같이 룬부스(Rhombus)기법을 이용하여, X세트의 주변에 있는 4개의 •세트 픽셀값의 평균값을 통해 구한다.

$$u_p = \lfloor \frac{v_1 + v_2 + v_3 + v_4}{4} \rfloor \quad (9)$$

$$d_{i,j} = u_{i,j} - u_p \quad (10)$$

(10)식에서 보는 것처럼 데이터를 은닉하는 과정은 Thodi와 Rodriguez의 기법과 같다. 데이터를 은닉하기 전에 먼저 정렬 기법을 사용해서 은닉 했을 때 왜곡이 적은 픽셀을 찾아야 한다. 정렬을 위해서 상관계수 식(11)을 먼저 계산한다.

$$c = \frac{1}{4} \cdot \sum_{i=1}^4 (v_i - \bar{v})^2 \quad (11)$$

이 때 상관계수인 c 값이 높다는 것은 데이터를 은닉했을 때 왜곡도가 높게 나타난다는 것이다. 따라서 정렬을 시킬 때는 c값이 낮은 순서대로 정렬을 시켜서 순서대로 데이터를 은닉해야 한다. cross set를 정렬시킨 후 데이터를 은닉시키고, 그 후에 dot set을 정렬시킨 후 데이터를 은닉한다. 디코딩하는 과정은 반대로 하게 되면, Thodi와 Rodriguez의 기법에서 구사할 수 없었던 정렬 기법을 사용할 수 있다.

Vasily는 데이터를 서로 겹치지 않는 부분인 cross와 dot으로 분리를 해서 데이터를 임베딩시키는 전략으로 Thodi와 Rodriguez의 기법에다가 정렬 기법을 사용할 수 있었다. 따라서 정렬 기법의 장점인 높은 PSNR을 얻을 수 있게 되었다.

5. 더블 임베딩

위에서 설명한 방법들은 최대한 1bpp의 은닉 용량을 가진다. 더블 임베딩은 Vasily가 새롭게 제안한 방법으로 이론상 2bpp까지 은닉 용량을 늘릴 수가 있다.

일반적으로 1bpp에서 2bpp로 용량을 늘리게 되면은 이미지 왜곡이 심해진다. Vasily는 이를 극복하기 위해서 예측값을 두 개를 이용해서 한 개의 픽셀에 2개의 데이터를 은닉시키게 된다.

먼저 두 개의 예측값중 하나를 택해서 데이터를 은닉시킨후 다른 예측값과 이미 데이터가 은닉된 픽셀의 밝기의 차분값을 이용해서 데이터를 은닉시킨다. 이렇게 두 번 데이터를 은닉하게 되면은 데이터 왜곡은 오히려 줄어들게 된다.

이 방법의 한계는 Kamstra가 제안한 정렬을 사용할 수 없다는 것이다. 데이터를 은닉시키는 과정에서 이미 변환된 값을 다시 이용해서 예측값을 구하는데 사용하기 때문에 정렬 순서를 바꾸게 되면은 은닉된 데이터를 추출하지 못하게 된다. 그러나 이 방법은 지금까지의 방법중중에서 가장 많은 주변 픽셀을 이용해서 예측값을 구하기 때문에 데이터 왜곡을 가장 최소로 잡을 수 있다.

III. 제 안

더블 임베딩 기법의 최대의 단점은 정렬을 사용할 수 없다는 것이다. 이를 극복하기 위해서 픽셀 주변의 8개의 점을 정사각형 모양으로 잡은 다음에 가장 자리만 겹치게 하는 방법을 택한다면, 정렬을 사용할 수 있다. 겹치는 가장 자리에는 데이터가 은닉되지 않기 때문에 정렬을 사용할 수가 있다. 그리고 더블 임베딩의 기법을 두 번 이용하게 되면 데이터 왜곡을 더욱 줄이거나 혹은 더블 임베딩했을 때와 거의 비슷한 왜곡도가 나올 것으로 기대된다.

감사의 글

본 연구는 지식경제부 및 한국 산업 기술 평가 관리원의 IT 산업원천기술개발사업의 일환으로 수행하였음. (2008-F-036-02, 익명성 기반의 u-지식 정보 보호기술개발)

[참고 문헌]

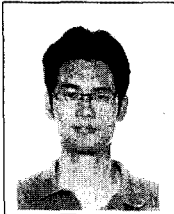
[1] P. G. Howard, F. Kossentini, B. Martins, S. Forchhammer, and W. J. Rucklidge, "The emerging JBIG2 standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 838-848, 1998.

[2] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890-896, 2003.

[3] L. H. J. Kamstra, A. M. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting," *IEEE Transactions on Image Processing*,

vol. 14, no. 12, pp. 2082-2090, 2005

- [4] D. M. Thodi, and J. J., Rodriguez, "Reversible watermarking by prediction-error expansion," *IEEE Southwest Symposium on Image Analysis and Interpretation*, Lake Tahoe, CA, pp. 21-25, 2004.
- [5] Vasilij Sachnev, Hyoung Joong Kim "Reversible Watermarking Algorithm Using Sorting and Prediction" *IEEE Transaction on Circuit System and Video Technology*. Accepted to publish.



이 병 용

2002년 건국대학교 수학과 졸업
2006년 고려대학교 수학과 석사 졸업
2006년~현재 고려대학교 정보경영공학전문
대학원 박사과정 재학중
<관심분야> Reversible Watermarking,
Divisible Load, Data Hiding

<e-mail> rom10910@korea.ac.kr

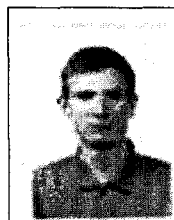


김 형 중

1978년 서울대학교 제어계측공학과 졸업
1986년 서울대학교 제어계측공학과(공학석사)
1989년 서울대학교 제어계측공학과(공학박사)
1990년~2006년 강원대학교 교수
2007년~현재 고려대학교 정보경영공학전문
대학원 교수

<관심분야> Watermarking, Parallel Computing, Image Hashing,
Data Compression, Steganography

<e-mail> khj-@korea.ac.k



Sachnev Vasilij

2002년 Komsomolna-na-Amure State
Technical University 전기공학과 졸업
2004년 Komsomolna-na-Amure State
Technical University 전기공학과 졸업
(공학석사)
2009년 고려대학교 정보경영공학전문대학원
졸업(공학박사)

2009년~현재 고려대학교 정보경영공학전문대학원 연구 교수

<관심분야> Reversible Watermarking, Steganography, Machine
Learning, Visible Reversible Watermarking

<e-mail> bassvasys@hotmail.com