

Histogram-Based Image Hash 성능 개선 방법

권하나* , 김소영* , 김형중*

*고려대학교 정보보호기술 연구센터(CIST)

e-mail : apu22@korea.ac.kr

An Improved Histogram-Based Image Hash Method

Hana Kwon*, SoYoung Kim*, HyoungJoong Kim*

*Center for Information Security Technologies, Korea Univ

요 약

Image hash는 영상에서 유사성을 찾는 방법으로 사용될 수 있는 기술자(Descriptor)로 특징지을 수 있다. 많은 image hash 방법중에 Histogram-based image hash는 Histogram equalization을 제외한 보통 잡음 및 다양한 기하학적 변조를 주어도 같은 그림을 찾아내는데 강력한 기능을 수행한다. 본 논문에서는 Histogram-Based Hash를 생성함에 있어 서로 다른 3개의 bin의 관계를 이용하여 Hash를 생성하였다. 본 논문은 이를 통해 영상의 유사성을 찾아내는데 있어 원본영상에 대해 기하학적 변조뿐만 아니라 상대적으로 성능이 약했던 Histogram equalization을 이용한 변조에 대해서도 성능이 개선되었다. 또한 가우시안 필터링의 알파 값을 다르게 지정함으로써 인하여 생성되는 두 히스토그램을 이용하여 기존의 방법보다 성능이 개선되었다.

1. 서론

Image hash는 오랫동안 연구자들에게 흥미있는 주제가 되어왔으며 많은 연구들이 있어 왔다.[1][2][3][5][7] Image hash는 content검색, image 검색, 정보은닉과 image 포렌식등 다양한 분야에서 인증(authentication)을 위한 방법으로 사용된다. 특히 histogram-based image hash는 기하학적 변형에 대한 image histogram 모양의 불변성을 이용함으로써 강력한 image hash알고리즘[6]으로 제안되어왔다. 이 방법은 hash의 강인함(robustness)를 향상시키기 위해 선행단계로 Gaussian filter를 적용하였으며 hash를 생성시 인접하는 두 bin그룹 사이의 pixel수에서의 관계를 비교하였다. 그러나 이 방법은 histogram equalization에 의한 변형에는 상대적으로 약한 결과를 나타냈다. 본 논문에서는 hash 생성시 각각 bin에

대해 양쪽에 이웃하는 bin의 관계를 비교하여 hash 값 생성을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서는 기존의 histogram-based image hash 생성방법과 제안된 방법을 살펴보고 3장에서는 두 방법간의 실험 결과를 비교하고 4장에서는 결론을 통해 본 논문을 맺는다.

2. Histogram-Based Image hash생성방법

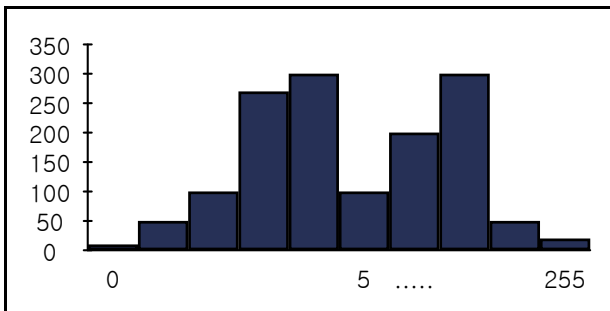
Image hash 생성은 먼저 선행단계를 거친 영상으로부터 특징을 추출하고 추출된 특징을 사용하여 hash값을 생성하는것이다. 선행단계는 변형에 대해 hash값이 강인함(robustness)를 갖을 수 있도록 하는 역할을 한다. 다음은 hash 생성 절차를 설명한다.

2.1 영상에 필터 적용

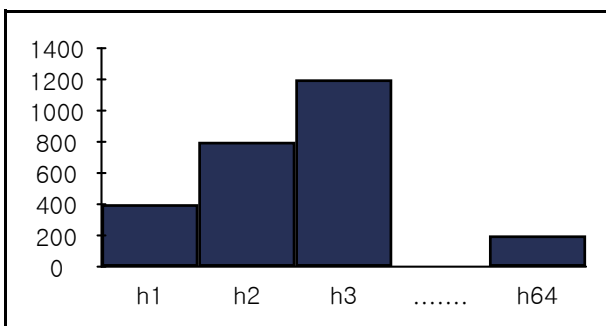
Hash의 강인성(robustness)를 보장하기 위한 방법으로 영상에 Low-pass filtering인 Gaussian filtering을 적용한다.

2.2 히스토그램 구하기

필터링 작업이 끝나면 필터링이 적용된 영상으로부터 히스토그램을 구할 수 있다. 흑백영상의 픽셀이 0부터 255의 값을 가지므로 히스토그램은 256 bin을 갖게 되고 된다. Hash 값을 생성하기 위해 256 bin을 L개로 그룹화 한다. 예를 들어 이웃하는 bin을 4개씩 그룹화하고자 하면 L은 '64'가 되고 8개씩 그룹화 하고자한다면 L은 '32'가 된다. 그림 1과 그림2의 히스토그램을 보면 쉽게 이해할 수 있다. 그림1은 각 픽셀에 해당하는 영상을 구성하는 원소를 개수를 나타낸 히스토그램이다. 그림2는 L=64로 h1부터 h64까지 히스토그램을 그룹화하였다. L개의 bin으로 이루어진 히스토그램을 추출할 수 있다.



(그림 1) 모든 pixel에 대한 영상 Histogram의 예



(그림 2) 64bins로 그룹화한 Histogram의 예

2.3 Hash값 구하기

두 개의 다른 bin그룹을 비교함으로써 Hash 값을 얻을 수 있다.

i번째 bin에서의 픽셀 수를 $h(i)$ 라고 하자. 조건 $1 \leq$

$i < L$ 과 $i < j \leq L$ 을 만족하는 서로 다른 bin을 $\{h(i), h(j)\}$ 과 같이 그룹화 해보자. 이 경우 그룹의 수는 $C_L^2 = L(L-1)/2$ 이 된다. 그룹 $\{h(i), h(j)\}$ 에 대해 $h(i)$ 과 $h(j)$ 을 다음 방법을 통해 hash bit 값을 나타낼 수 있다

$$bit = \begin{cases} 1 & \text{if } h(i)/h(j) \geq 1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

2.4 Hamming distance 계산

Hash 값을 일차원 벡터값 $hash(k)$ 라 하면 ($1 \leq k \leq C_L^2$)한쌍의 hash에 대한 이들간의 Hamming distance값은 다음과 같이 정의할 수 있다.

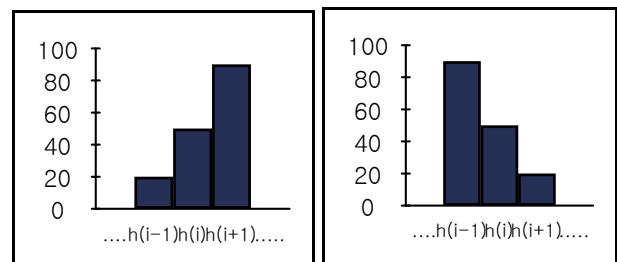
$$d(hash_1, hash_2) = \frac{1}{C_L^2} \sum_{k=1}^{C_L^2} |hash_1(k) - hash_2(k)| \quad (2)$$

이때 hamming distance값이 0에 가까우면 비슷한 영상, 0.5에 가까우면 다른 영상이라고 말할 수 있다.

3. 제안된 방법

(1) 서로 다른 세 개의 bin을 이용하는 방법

기존의 방법에서 두 개의 다른 bin그룹을 비교하여 Hash 값을 구하는 방법을 인접한 세 개의 bin그룹의 관계를 비교함으로써 Hash값을 구하는 방법을 이용하였다. 그림3과 같이 세 개의 이웃하는 bin그룹간의 관계가 계단식의 증가나 감소의 모양을 갖고 있으면 '1'값을 주고 그 외는 '0'값을 주어 Hash값을 생성하였다.



(그림 3) hash값 '1'을 갖는 세 개 bin의 히스토그램 모양

즉,

$$bit = \begin{cases} 1 & \text{if } h(i-1) \leq h(i) \leq h(i+1) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

로 정의한다.

이렇게 세 개의 bin그룹의 관계를 연관지음으로써 두 개의 bin에 대해서만 비교할때보다 hash값에 영상에 대한 원 정보가 더 강력하게 포함될 수 있다.

(2) Gaussian filtering 알파 값에 따른 해쉬 생성

이 방법은 해쉬 이미지 해쉬 생성에 대한 새로운 방법을 제시한다. 기존의 이미지 히스토그램 방법에서 있었던 전처리 과정 중 Gaussian filtering에 해당하는 부분으로 알파 값에 따라서 히스토그램이 변화한다는 데에 기반을 두었다. 히스토그램은 같은 이미지 라면 기하학적인 변화에 크게 변화하지 않는다. 이에 따라 Gaussian filtering의 알파값에 변화를 주면 위 아래로 높이의 변화는 있지만 전체적인 변화에는 영향을 주지 않기 때문에 이미지 검색 기능에 탁월하다.

해쉬 생성방법은 다음과 같다.

$$bit = \begin{cases} 0 & \text{if } (hist_{0.5}(i) < hist_{ori}(i)) \ \& \ (hist_{3.0}(i) > hist_{ori}(i)) \\ 1 & \text{otherwise} \end{cases}$$

이 방법은 알파값을 0.5로 하였을 경우와 원본이미지와 알파값을 3.0으로 하였을 경우 각각의 히스토그램을 비교하여 해쉬를 생성한 후 비교하는 방법이다.

4. 실험 결과

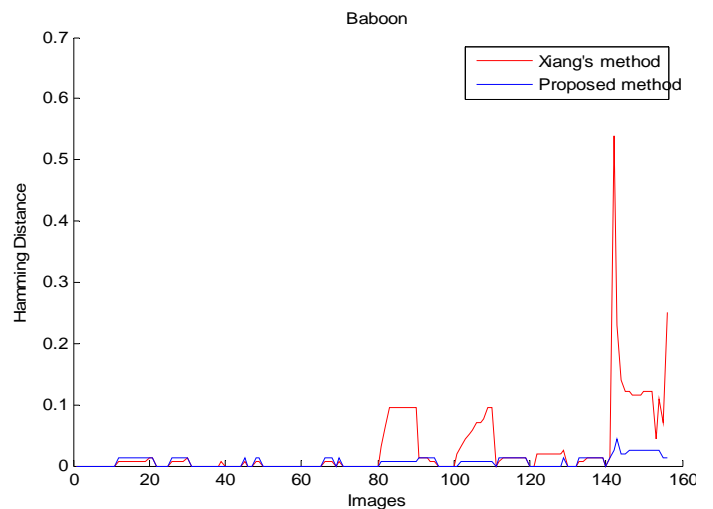
실험을 통해 기존의 hash 생성방법과 제안된 방법을 비교 분석해보았다. 실험방법은 한 영상에 대해 어떤 변형이 이루어졌을때 변형이 이루어진 영상이 원본영상이 변형된 것이라고 판단할 수 있는지에 대한 강인성(Robustness)실험과 서로 다른 영상을 비교했을때 다른 영상이라고 판단할 수 있는 유일성(Uniqueness)실험을 실시하였다.

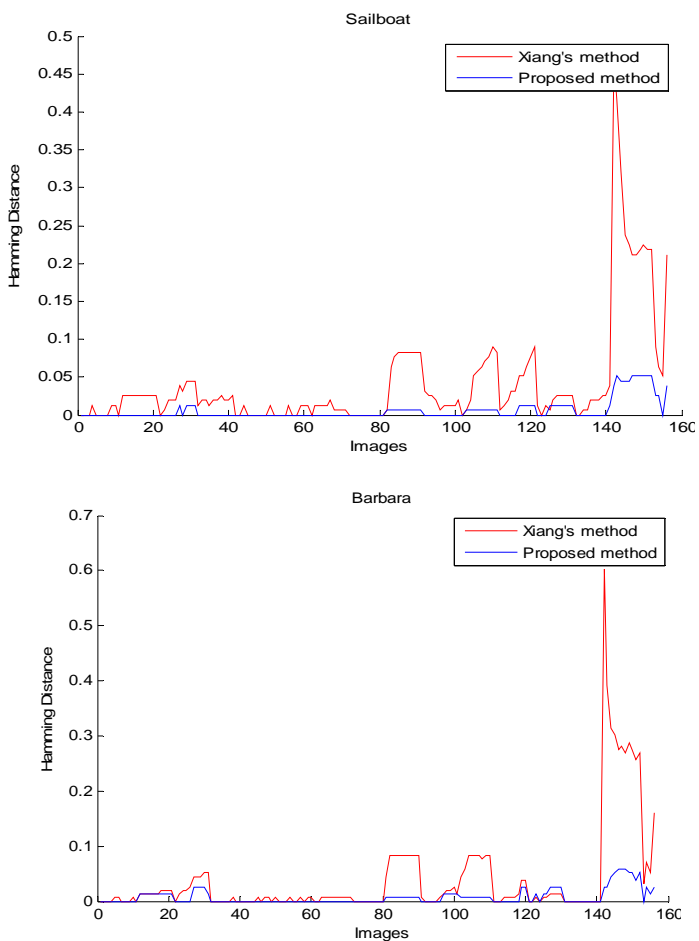
4.1 강인성(Robustness) 실험

Histogram based image hash 방법은 회전등 기하학적 변형에 강인성을 보이지만 histogram equalization 변형에 대해서는 충분한 스키마가 되지 못했다. 그러나 제안된 세 개의 bin의 관계를 이용해 생성한 hash값의 비교방법은 histo equalization 변형 및 text를 덧씌운 변형에도 기존의 방법보다 우수한 결과를 보였다.

실험은 압축, 필터링, 회전등 변형을 준 155개의 복제 영상을 이용하여 실시하였다. 실험에 사용된 원본 영상은 4가지 이미지(Sailboat, Lena, Baboon, Barbara)이며 원본을 변형하기 위해 사용된 방법은 총 155개로 영상1-10(압축률은 달리한 압축), 11-20(quality율을 달리한 압축), 21-40(Gaussian과 Uniform 잡음 적용), 41-70(Gaussian filter, Median filter, With filter적용), 71-80(회전), 81-90(크기변환), 91-100(shearing), 101-110(crop), 111-140(굴곡, 떨림), 141-151(histogram equalization), 152-155(text 덧입힘) 변형을 적용하였다.

기존 image hash와 제안된 image hash를 이용하여 원본영상과 변형이 적용된 각각의 155개 영상과의 hamming distance를 구한 결과는 다음과 같다.





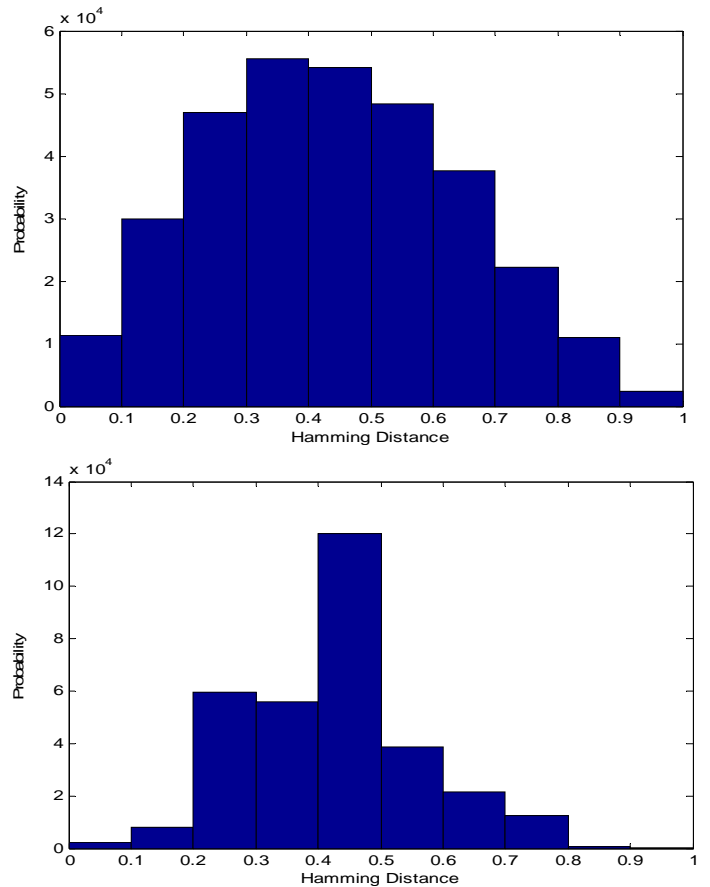
(그림 4) 기존방법과 제안된 방법을 이용한 원본영상과 변형 영상의 Hamming distance 비교 (위로부터 Baboon, Sailboat, Barbara 영상)

그림 4에서 볼 수 있듯이 기존의 image hash 생성 방법은 크기변환과 자르기(Cropping)변환에 대해 Hamming distance 값이 0.1가량 되었고 특히 histogram equalization 과 text변형에 대해서는 다른 변형에 비해 0.5 이상이 되는 등 다소 높게 나타났다. 반면 세 개의 bins의 관계를 이용하여 hash를 생성한 제안된 방법은 모든 변형에 대해 원본과의 hamming distance 값이 0.05가량으로 '0'에 가깝게 일정함 보였다.

4.2 유일성(Uniqueness) 실험

서로 다른 영상중 랜덤하게 두 영상을 선택할 때 각각의 영상이 서로 다른지를 얼마나 잘 판단할 수 있는지에 대한 실험이 유일성(Uniqueness) 실험이다. 본 실험에는 총 800개의 흑백영상을 이용하였다. 그

결과는 그림 5와 같다.

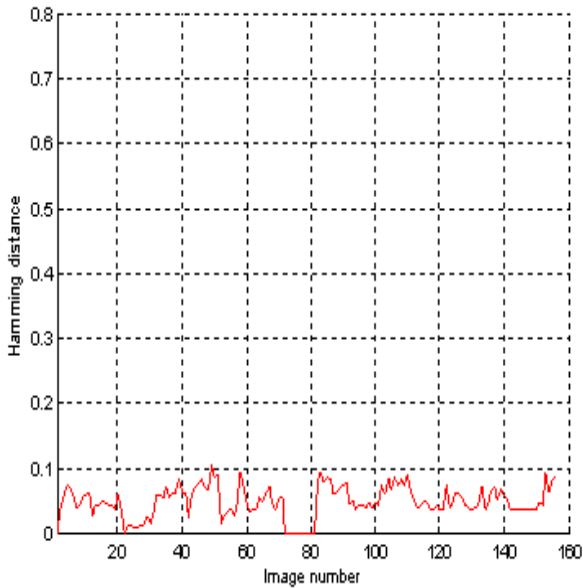


(그림 5) 800개 영상중 두 개의 영상을 선택해서 구한 Hamming distance 분포(위로부터 기존 방법, 제안된 방법)

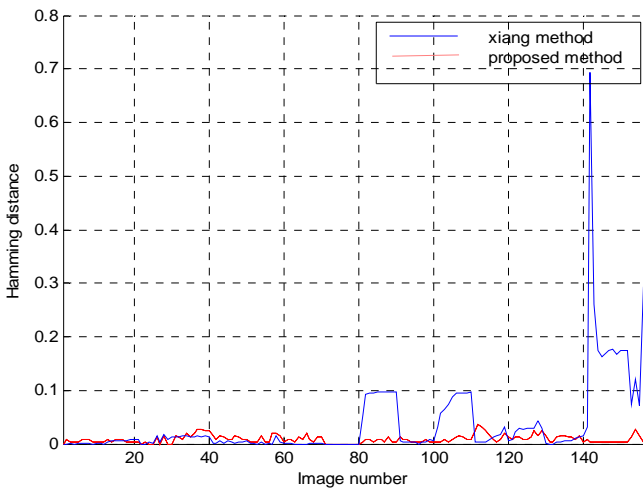
실험결과, 기존방법의 Hamming distance 평균값은 0.4419 분산은 0.0406이며 제안된 방법의 평균값은 0.4392 분산은 0.0190으로 거의 비슷한 결과를 보였으며 분산값으로 볼때 제안된 방법의 측정 Hamming distance가 평균값에 가깝게 분포되어 있음을 알 수 있다.

4.3 Gaussian filtering 알파값에 대한 실험

이 실험 또한 156개의 공격 이미지를 가지고 실험하였다. 이 방법은 기존의 방법에서 약했던 histogram equalization 과 text변형에 대해서 모두 Hamming distance 값이 0.1 이하로 나옴으로써 성능이 개선되었음을 알 수 있다.



(그림 6) Lena 이미지를 이용한 제안된 방법의 Hamming distance 비교



(그림 7) Baboon 이미지를 이용한 기존방법과 제안된 방법의 Hamming distance 비교

5. 결론

본 논문에서는 매우 간단하고 쉬운 방법으로 개선된 Histogram based image hash를 제안하였다. Hash를 생성하는 방법에서 추출된 히스토그램의 세 개 bin의 관계를 이용함으로써 기존 두 개 bin의 관계를 이용한 방법보다 만족스러운 결과를 얻었다. 또

한 Gaussian filtering의 알파값의 변화에 따른 방법 또한 기존의 방법보다 개선되었음을 알 수 있었다. 특히 제안된 방법으로 Histogram equalization 변형에 취약했던 기존 방법이 상당부분 보완되었다.

[참고문헌]

- [1] E. Y. Chang, C. Li, J. Z. Wang, P. Mork, and G. Wiederhold . "Searching near replicas of images via clustering." In Proceedings of SPIE Multimedia Storage and Archiving Systems VI, September 1999
- [2] Y. Ke, R. Sukthankar, and L. Huston. "An efficient partsbased near-duplicate and sub-image retrieval system." In Proceedings of the ACM International Conference on Multimedia, October 2004
- [3] N. Shivakumar and H. Garcia-Molina. "Finding near-replicas of documents and servers on the web." In Proceedings of Workshop on Web Databases, February 1998
- [4] A. Swaminathan, Y. Mao, M. Wu. "Robust and secure image hashing." IEEE Transactions on Information Forensics and Security, Vol.1, no.2, June 2006
- [5] B. Wang, Z. Li, M. Li, and W.-Y. Ma, "Large-scale duplicate detection for Web image search." IEEE International Conference on Multimedia and Expo, July 2006
- [6] S. Xiang, H. J. Kim, and J. Huang. "Histogram-based image hashing scheme robust against geometric deformations." In Proceedings of the ACM Multimedia and Security Workshop, September 2007
- [7] D. Q. Zhang and S.-F. Chang. "Detecting image near-duplicate by stochastic attributed relational graph matching with learning." In Proceedings of the ACM Multimedia, October 2004