The original publication is available at
www.springerlink.com .

Image Distortion Estimation by Hash Comparison

Li Weng and Bart Preneel*

Katholieke Universiteit Leuven, ESAT/COSIC-IBBT

`li.weng@esat.kuleuven.be, bart.preneel@esat.kuleuven.be`

Abstract. Perceptual hashing is conventionally used for content identification and authentication. In this work, we explore a new application of image hashing techniques. By comparing the hash values of original images and their compressed versions, we are able to estimate the distortion level. A particular image hash algorithm is proposed for this application. The distortion level is measured by the signal to noise ratio (SNR). It is estimated from the bit error rate (BER) of hash values. The estimation performance is evaluated by experiments. The JPEG, JPEG2000 compression, and additive white Gaussian noise are considered. We show that a theoretical model does not work well in practice. In order to improve estimation accuracy, we introduce a correction term in the theoretical model. We find that the correction term is highly correlated to the BER and the uncorrected SNR. Therefore it can be predicted using a linear model. A new estimation procedure is defined accordingly. New experiment results are much improved.

1 Introduction

Perceptual hash (PH) algorithms are a particular category of hash functions for the multimedia domain [1–3]. They generate hash values based on robust features of multimedia data. A hash value is typically a compact binary string. Differing from cryptographic hash values [4, 5], a PH value is only based on the content. It is usually insensitive to incidental distortion, such as compression, scaling, and photometric processing. Therefore it can be used as a persistent fingerprint of the corresponding content. One can efficiently find similar multimedia content by comparing PH values. Therefore, PH algorithms are useful tools for applications such as multimedia databases and multimedia search engines.

* This work was supported in part by the Concerted Research Action (GOA) AM-BioRICS 2005/11 of the Flemish Government and by the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy). The first author was supported by the IBBT/AQUA project. IBBT (Interdisciplinary Institute for Broad-Band Technology) is a research institute founded in 2004 by the Flemish Government, and the involved companies and institutions (Philips, IPGlobalnet, Vitalys, Landsbond onafhankelijke ziekenfondsen, UZ-Gent). Additional support was provided by the FWO (Fonds Wetenschappelijk Onderzoek) within the project G.0206.08 Perceptual Hashing and Semi-fragile Watermarking.

The usage of perceptual hashing has been evolving. It is usually used for multimedia content identification [1]. It is also used for generating content-based watermarks [2]. New PH algorithms are sometimes designed to support a secret key [3]. The hash value being highly dependent on the key enables message authentication. Compared with conventional message authentication codes (MAC) or digital signatures (DS) [4, 5], *perceptual* MAC and DS do not need to be re-generated when the multimedia data undergoes incidental distortion. Some PH algorithms also support tamper localization [6–8]. It seems that PH algorithms are having growing potential for new applications.

Recently, a new application of perceptual hashing has been found. Besides content identification and authentication, perceptual hashing can also help with quality assessment or distortion estimation. Since the hash value is related to the content, quality degradation might reflect from the hash value. Given a multimedia file and its distorted (compressed) version, one can tell the distortion level by comparing their PH values. This idea was successfully demonstrated by Doets and Lagendijk for estimating music compression distortion [9]. An interesting question is whether the same idea works for images or video. In this work, we use an image hash algorithm for image distortion estimation. We show that it is possible to estimate the signal to noise ratio according to the hash distance. The considered distortions are the JPEG compression, the JPEG2000 compression, and the additive white Gaussian noise (AWGN). The rest of the work is organized as follows: Section 2 introduces the background of image distortion estimation; Section 3 proposes an image hash algorithm, and introduces a model for distortion estimation; Section 4 shows experiment results; Section 5 proposes a method to improve estimation accuracy; Section 6 concludes the work.

2 Image Distortion Estimation

Image distortion estimation is a conventional research topic. It is closely related to image quality assessment and image tamper detection/localization. There have been many approaches which generally divide into two categories – ones that require the original data, e.g., [10, 11], and ones that do not. The former is generally more accurate, while the latter is more practical (since original data is not always available). Recently, it was found that hash comparison can also indicate the distortion to some extent [9] – a new feature offered by perceptual hashing. This approach is essentially a trade-off between the above two categories. Since a hash value is related to the content, distortion also influences the hash value. A typical user case is the distribution chain of high quality multimedia content. In a heterogeneous network, the quality of content generally degrades from the source/server (e.g. a production studio) to the end-user/client (e.g. a mobile device). As a content provider, it is often necessary to monitor the quality (degradation) along the distribution chain. Although parameters such as the bit rate can tell the quality to some extent, they are generally not reliable. It is better to assess the quality on the client side and report to the server. Due to many reasons, e.g., bandwidth or copyright limit, original multimedia data

cannot be available to the end-users. Instead, sending hash values is a good alternative. End-users could compute hash values from the received content and compare them with the original ones to estimate the quality. Figure 1 illustrates this idea. Besides distortion estimation, the availability of hash values also makes it easy to perform content-based search or content authentication.

Fig. 1. An example application scenario of distortion estimation.

More specifically, distortion estimation by hash comparison also has two kinds of approaches. Some only use the hash values; some also make use of the received multimedia data. Intuitively, utilizing both the hash value and the distorted version gives the best results, because more information is available. Tagliasacchi *et al.* have already demonstrated this approach [7]. They consider an optimization problem – given the original hash and the distorted signal, search for a sparse error pattern which results in the distorted hash. However, it is computationally expensive and requires very sophisticated algorithms. In this work, we only exploit hash values. This method is simple and fast. We focus on the estimation of compression distortion. The signal to noise ratio (SNR) is used as the distortion metric. In the following, we first propose an image hash algorithm.

3 An Image Hash Algorithm for Distortion Estimation

We have been inspired by the work of Doets and Lagendijk [9], where they estimate music compression distortion by comparing audio hash values. They used the audio hash algorithm proposed by Haitsma and Kalker [12]. Although it was designed for audio, we find that the basic principles might also apply to images. It is interesting to see how well it works in a different domain. Therefore, we adapted the original audio hash algorithm for images.

The new algorithm is particularly designed for high definition (HD) images, because distortion estimation is more interesting for such content. The algorithm is block-based. A sub-hash value is computed from each block. There are the following steps:

- Pre-processing: the input image is converted to gray-scale and proportionally resized so that the maximum length is 1280 pixels; the mean value is subtracted from the image;

- Block division: the pre-processed image is first padded then divided into 64×64 pixel blocks with 50% overlapping;
- Power spectral density (PSD) estimation: the PSD of block n is estimated from the 2D Fourier transform of the corresponding image block $X_n(k)$:

$$S_n(k) = \frac{1}{L^2} |X_n(k)|^2, \quad L = 64; \quad (1)$$

- PSD extraction: for each block PSD, the low frequency components corresponding to normalized frequencies 0–0.5 are extracted, see Fig. 2;
- Frequency band division: each extracted (2D) PSD is divided into frequency bands of size 16×16 ; the energy of band m is computed by summing up all values in the band:

$$E_{n,m}^b = \sum_{k \text{ in band } m} S_n(k); \quad (2)$$

- Compute difference between frequency bands in each block:

$$ED'_{n,m} = E_{n,m}^b - E_{n,m+1}^b; \quad (3)$$

- Compute difference between blocks:

$$ED''_{n,m} = ED'_{n,m} - ED'_{n+1,m}; \quad (4)$$

- Hash generation:

$$H(n, m) = \begin{cases} 1, & \text{if } ED''_{n,m} > 0 \\ 0, & \text{if } ED''_{n,m} \leq 0 \end{cases}. \quad (5)$$

The final hash value is the concatenation of all sub-hash values.

The canonical size 1280 in the algorithm is relatively large compared with conventional values such as 512. This is because a small canonical size makes the hash value insensitive to slight distortion introduced by compression. In order to suit the particular need of the application, the hash is designed to have slightly less robustness. Since the pre-processing keeps the original aspect ratio, the size of a hash value is not constant. The maximum size is 4563 bits for the case of 1280×1280 pixels. Although it is larger than conventional ones which are below 1000 bits, it is still negligible considering the data volume of HD content.

3.1 SNR Estimation

A model was derived in [9] to describe the relationship between the SNR and the BER. It also turns out to work for the above algorithm. In this section, we briefly introduce this model. Denote the original image signal by x and the distorted one by y . The distortion is modeled as additive noise, denoted by w :

$$y(i) = x(i) + w(i). \quad (6)$$

Following the notation in the previous section, denote the block difference by ED''_X and ED''_Y respectively. In the derivation of the model, the following assumptions are made:

Fig. 2. Frequency band extraction and division.

- *Assumption 1:* The signal contribution ED_X'' and the noise contribution $ED_Y'' - ED_X''$ are zero-mean, mutually independent, normally distributed random variables;
- *Assumption 2:* $x(i)$ and $w(i)$ are zero-mean, normally-distributed random variables with variance σ_X^2 and σ_W^2 .

According to Assumption 1 [9, Appendix 1], we have:

$$P_e = \frac{1}{\pi} \left(\sqrt{\frac{\text{VAR}[ED_Y'' - ED_X'']}{\text{VAR}[ED_X'']}} \right). \quad (7)$$

Assumption 2 gives the following relationship [9, Appendix 2-3]:

$$\text{VAR}[ED_Y'' - ED_X''] = \left(\frac{\sigma_W^4}{\sigma_X^4} + 2 \frac{\sigma_W^2}{\sigma_X^2} \right) \text{VAR}[ED_X'']. \quad (8)$$

Defining $\text{SNR} = \frac{\sigma_X^2}{\sigma_W^2}$, we have the relationship between the BER P_e and SNR by combining (7) and (8):

$$P_e = \frac{1}{\pi} \arctan \left(\sqrt{\frac{\sigma_W^4}{\sigma_X^4} + 2 \frac{\sigma_W^2}{\sigma_X^2}} \right) = \frac{1}{\pi} \arctan \left(\sqrt{\frac{1}{\text{SNR}^2} + \frac{2}{\text{SNR}}} \right). \quad (9)$$

Knowing the BER, solving the above equation is straight-forward:

$$\text{SNR} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \quad (a = \tan(\pi P_e)^2, b = -2, c = -1). \quad (10)$$

There is only one meaningful solution.

Equation (10) is a convenient tool to estimate the SNR by BER. Figure 3 shows the ideal relationship. The asterisks in the figure are the BER values computed from synthesized data. In the simulation, we first set a specific SNR, then randomly generate 1280×1280 input images and add noise according to the SNR. Both the input signal and noise are normally distributed with zero mean. For each SNR, we repeat 10 times and plot the average BER value. Clearly, the computed BER values fit the theoretical curve perfectly. Therefore, the proposed algorithm is theoretically valid. In reality, the relationship deviates from the ideal situation. It is reported in [9] that the errors can be on the same order as the SNR values. It is not known yet to which extent the same methodology could work for images. In the next section we show experiment results with real data.

Fig. 3. Ideal relationship between SNR and BER.

4 Experiment Results

The proposed algorithm has been tested using 900 gray-scale natural-scene photos. The results can also extend to color images. Their content includes architecture, art, industry, mechanics, animals, humanoids, landscapes, vehicles, etc. The image size varies from 960×1280 to 1920×2560 pixels. We focus on the algorithm’s capability of distortion estimation. Three kinds of distortions are considered – the JPEG compression, the JPEG2000 compression, and additive white Gaussian noise. They are commonly encountered in practice. The distortion is measured by SNR. However, we use a slightly modified definition of SNR – the mean value of the original image is subtracted from both the original signal x and the distorted signal y before calculation:

$$\text{SNR}[x, y] = \text{SNR}[x - \text{mean}(x), y - \text{mean}(x)] . \quad (11)$$

In this way, we meet one of the assumptions that the input signal is zero-mean; the noise signal $y - x$ is still the same. This definition allows better estimation accuracy. In the following, we always refer to the zero-mean SNR. We evaluate the estimation performance by the error ratio

$$\text{error ratio} = \frac{|\text{SNR}_{\text{real}} - \text{SNR}_{\text{estimated}}|}{\text{SNR}_{\text{real}}} \times 100\% . \quad (12)$$

The SNR values are always expressed in dB.

For each original image in the test database, a few distorted versions are generated by the operations listed in Table 1. For each pair of original and distorted images, their hash values are computed and compared; the real SNR is compared with the estimated SNR, which is computed by inputting the BER to (10). Figure 4 shows the SNR values versus the estimated ones. The x-axis is the real SNR, and the y-axis is the estimated SNR. Ideally, the estimated values (red crosses) should lie on the (blue dash-dotted) diagonal line. The spread of the “clouds” indicates that the estimation errors are relatively large. The average

Table 1. Distortions to be estimated.

Distortion	Parameter range (step)	Error ratio	Improved error ratio
JPEG	Quality Factor: 30–90 (10)	28.64%	3.96%
JPEG2000	Compression ratio: 0.3–0.9 (0.1)	22.33%	2.70%
AWGN	SNR: 20 dB–50 dB (5)	31.56%	17.04%

error ratios (22.33%–31.56%) are listed in the 3rd column of Table 1. They probably restrict the application to the classification among a few SNR levels.

The errors are essentially due to the non-Gaussianity and the dependence between signal and noise. Large estimation variance was also observed in [9]. The authors discussed some possible ways to reduce the variance, such as discarding unreliable hash bits, averaging across space/time. These approaches may not be suitable for images due to the limited data. In the next section, we propose our own method to improve estimation accuracy.

5 Improve Estimation Accuracy

The large estimation errors in the previous tests show that the assumptions in Section 3.1 do not hold for natural-scene images. Equation (9) is not exact in practice. Therefore, a better model is required. However, for the same SNR, both the BER and the estimated SNR exhibit large variance. That means a single model (curve) is not sufficient to describe the SNR-BER relationship for all images. Instead, a model should be derived for each image.

We assume that in practice (9) requires a correction term C :

$$P'_e = \frac{1}{\pi} \arctan \left(\sqrt{\frac{1}{\text{SNR}^2} + \frac{2}{\text{SNR}} + C} \right). \quad (13)$$

Correspondingly, (10) becomes:

$$\text{SNR} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \quad (a = \tan(\pi P_e)^2 - C, b = -2, c = -1). \quad (14)$$

When estimating the SNR, best accuracy can be achieved if the correction term is available. However, storing all correction terms is certainly not a good option. A much more practical way is to predict the correction term from existing knowledge. Denote the SNR value estimated without any correction by SNR_0 . By observing the correction terms for all the distorted images in the previous section, we find that, for one image, the correction term for a specific SNR is likely to be correlated to the corresponding BER and SNR_0 . Recall that, for each kind of distortion, 7 distorted images are generated for each original image. The vector of correction terms exhibits high correlation with the corresponding BER and SNR_0 vectors. Figure 5 shows the histogram of the absolute values of the correlation coefficients for JPEG. Most absolute correlation coefficients are close to 1. We observe the same situation for JPEG2000 and AWGN.

(a) JPEG: Average error ratio=28.64%.

(b) JPEG2000: Average error ratio=22.33%.

(c) AWGN: Average error ratio=31.56%.

Fig. 4. Real SNR vs. estimated SNR.

Fig. 5. Histogram of absolute correlation coefficients 1) between the correction term and the BER; 2) between the correction term and the SNR_0 for JPEG.

Based on the above observation, we assume there is a linear relationship between the correction term and the BER, SNR_0 values for a certain image:

$$C = a \cdot \text{BER} + b \cdot \text{SNR}_0 + c. \quad (15)$$

Knowing C , BER and SNR_0 , the weights a, b, c can be derived by solving linear equations. We use the data in the previous experiment and compute a, b, c for each original image by the least-square method. The experiment is then run once again. The SNR estimation now consists of three steps: 1) compute SNR_0 by (9); 2) compute the correction term by (15); 3) compute SNR by (14). Figure 6 shows the new estimation results. The clouds are much more compact. The average error ratios are listed in the last column of Table 1. There is significant improvement for all cases. The error ratios for JPEG and JPEG2000 are reduced to less than 4%. The error ratio for AWGN is almost halved. Therefore, the proposed method is effective.

In the above approach, we only used one correction term. More correction terms are likely to provide better estimation. The price to pay is that we need to store the model parameters a, b, c for each original image, and their derivation requires some training data. There is a trade-off between accuracy, storage, and pre-computing.

6 Conclusion

In this work, we explore a new application offered by image hashing techniques. By comparing the hash values of original images and their compressed versions, we are able to estimate the distortion level. A particular image hash algorithm is proposed for this application. The distortion level is measured by the signal to noise ratio (SNR). It is estimated from the bit error rate (BER) of hash values.

(a) JPEG: Average error ratio=3.96%.

(b) JPEG2000: Average error ratio=2.70%.

(c) AWGN: Average error ratio=17.04%.

Fig. 6. Real SNR vs. estimated SNR (improved).

The estimation performance is evaluated by experiments. The JPEG, JPEG2000 compression, and additive white Gaussian noise are considered. We show that a theoretical model does not work well in practice. The average error ratios are 28.64%, 22.33%, and 31.56% for the three kinds of distortion respectively. In order to improve estimation accuracy, we introduce a correction term in the theoretical model. We find that the correction term is highly correlated to the BER and the uncorrected SNR. Therefore it can be predicted using a linear model. A new estimation procedure is defined accordingly. New experiment results are much improved. The error ratios are reduced to 3.96%, 2.70%, and 17.04%.

References

1. Schneider, M., Chang, S.F.: A robust content based digital signature for image authentication. In: Proc. of International Conference on Image Processing (ICIP96). Volume 3. (1996) 227–230
2. Fridrich, J.: Robust bit extraction from images. In: Proc. of IEEE International Conference on Multimedia Computing and Systems. Volume 2. (1999) 536–540
3. Swaminathan, A., Mao, Y., Wu, M.: Robust and secure image hashing. IEEE Transactions on Information Forensics and Security **1**(2) (June 2006) 215–230
4. Schneier, B.: Applied Cryptography: Protocols, Algorithms, and Source Code in C. 2nd edn. John Wiley & Sons (1996)
5. Menezes, A., van Oorschot, P., Vanstone, S.: Handbook of Applied Cryptography. CRC Press (1996)
6. Roy, S., Sun, Q.: Robust hash for detecting and localizing image tampering. In: Proc. of International Conference on Image Processing. Volume 6. (2007) 117–120
7. Tagliasacchi, M., Valenzise, G., Tubaro, S.: Hash-based identification of sparse image tampering. IEEE Transactions on Image Processing **18**(11) (November 2009) 2491–2504
8. Lu, W., L. Varna, A., Wu, M.: Forensic hash for multimedia information. In: Proc. of SPIE Media Forensics and Security Conference. (2010)
9. Jan O. Doets, P., L. Lagendijk, R.: Distortion estimation in compressed music using only audio fingerprints. IEEE Transactions on Information Forensics and Security **16**(2) (February 2008) 302–317
10. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4) (April 2004) 600–612
11. Brooks, A., Pappas, T.: Using structural similarity quality metrics to evaluate image compression techniques. In: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2007. Volume 1. (15–20 April 2007) I–873–I–876
12. Haitisma, J., Kalker, T.: A highly robust audio fingerprinting system. In: Proc. of 3rd International Conference on Music Information Retrieval. (October 2002) 107–115