A Probabilistic Dissimilarity Matching for the DFT-Domain Image Hashing

Jin S. Seo*1, Myung-Suk Jo1

1Gangneung-Wonju National University, Gangneung, Korea
{jsseo*, msjo}@gwnu.ac.kr

Abstract
An image hash, a discriminative and robust summary of an image, should be robust against quality-preserving signal processing steps, while being pairwise independent for perceptually different inputs. In order to improve the hash matching performance, this paper proposes a probabilistic dissimilarity matching. Instead of extracting the binary hash from the query image, we compute the probability that the intermediate hash vector of the query image belongs to each quantization bin, which is referred to as soft quantization binning. The probability is used as a weight in comparing the binary hash of the query with that stored in a database. A performance evaluation over sets of image distortions shows that the proposed probabilistic matching method effectively improves the hash matching performance as compared with the conventional Hamming distance.

Keywords: image hashing, image identification, image fingerprinting, probabilistic dissimilarity.

1. Introduction
With the huge volume of images available for protection, browsing, and indexing, there is a strong need to identify a given image fast and reliably using its own features, which are called hash of the image [1]. Images often undergo various manipulations during distribution including compression, enhancement, geometrical distortions and analog-to-digital conversion that may preserve perceptual value. For example, humans do not readily perceive a perceptual difference between an original image and a moderately resized or compressed version. Thus the cryptographic hash functions [2], which map arbitrary length data to a small and fixed number of bits, cannot be employed for image hashing. The aim of image hashing is to make the hashes of the perceptually similar images as close as possible in a metric space. The basic requirements of an image hashing system are robustness, pairwise independence, and database (DB) search efficiency [1,3].

• Robustness (Invariance under perceptual similarity): The hash of a degraded image should be the same as or similar to the hash of the original image. Image hashing function should be sufficiently invariant with respect to the perceptual-quality preserving degradations.
• Pairwise independence (Collision free): Two images, that are perceptually different, must have
sufficiently different hash values. Hashes should be statistically independent or uncorrelated with each other to prevent different images from possessing similar hash values by chance.

• Database search efficiency: The structure of hashes must be conducive to fast database search. In practice, fast database search is essential for the commercial applications requiring real-time response.

The importance of each requirement depends on the intended applications, which includes filtering for file-sharing services, automated monitoring for broadcasting stations, commercial verification, duplicate file detection, and automated indexing of large-scale media archives. In general, improving the realization of one requirement usually results in degrading that of the other requirements. In the design of the hash function for each application, the different requirements should be traded off.

![Functional diagram of the image hashing system](image)

**Figure 1. Overview of the hash DB construction and matching.**

Functional diagram of the image hashing system is shown in Figure 1. To construct an image identification system, we first extract hashes from a number of images and store them in a hash DB. The hash extraction process comprises two steps: the intermediate hash extraction and the quantization, also referred to as binarization. In most cases, the intermediate hash of an image is given by a real-valued vector. The hash of an unknown query image is used for identification over the constructed hash DB. The candidates for the query hash are mostly obtained by the nearest neighbor DB search. To these candidates, the hash matching is applied for verification. Conventional hash matching compares the hash bits of the query image with the hash bits stored in the DB. This paper improves the hash matching performance by using the probabilistic dissimilarity matching, where probabilistic match scores are computed by the probability that the intermediate hash vector of the query image belongs to each quantization bin. We compute the probability that the intermediate hash vector of the query image belongs to each quantization bin, which is referred to as soft quantization binning. This probability value is used for deciding whether the query image corresponds to the quantized hash in the DB. We apply the proposed weighted hash matching to the widely-used DFT-domain image hashing. The performance of the proposed matching method was experimentally evaluated using thousands of images having various types of distortion, which verified that the proposed probabilistic matching method is effective for improving the hash matching performance as compared with the conventional Hamming distance.
This paper is organized as follows. In Section 2, the polar DFT image hashing method is described. In Section 3, we propose a probabilistically weighted distance measure for the transform-domain image hashing. In Section 4, the performance evaluation of the proposed matching method is presented. In Section 5, the performance and the limitations of the proposed method are summarized.

2. Polar DFT image hashing

A number of image hashing methods have been proposed. While spatial-domain hash extraction [4,5] is computationally simple and discriminant, transform-domain hash extraction [6-9], using the Radon, Fourier-Mellin, or Fourier transform, is conducive to making the resulting hash robust against geometric distortions. Among many transform-domain hashing methods, we chose the one based on the polar Fourier transform [7], because of its implementation simplicity. We note that the proposed hash matching method in Section 3 can be applicable to any kind of transform-domain image hashing with some reasonable modification. In [7], the polar coordinate over the two-dimensional Fourier transform was used for extracting a robust image hash. Let the polar Fourier transform of the given image be \( F(\rho, \theta) \). Along the \( \rho \) axis, we sample 64 circles with \( \rho_0, \rho_1, \ldots, \rho_{63} \), as shown in Figure 2. Along the \( \theta \) axis, we sample \( L \) equidistant points in the range of \([0, 2\pi]\) as in [7], typically \( L = 360 \). The intermediate hash \( H_I[j] \) of the \( j \)-th circle of an image \( I \) is given as a weighted sum along the \( \theta \) axis by

\[
H_I[j] = \sum_{i=0}^{L-1} \beta_{\rho_j,i} F(\rho_j, \frac{(2i+1)\pi}{L})
\]

where \( \beta_{\rho_j,i} \) are key-dependent pseudo random numbers, which are normally distributed with mean zero and unit variance. The obtained intermediate hash vector \( H_I \) is quantized by using the 6-bit gray code to form 64 (number of circles) by 6 hash bits, that is, 384 bits per image. In [7], the Hamming distance was utilized for comparing the hash of the query with the hashes in the DB for verification, as shown in Figure 1. Two images are declared similar if the Hamming distance between their hashes, usually expressed as the Bit Error Rate (BER), is below a certain threshold \( T \), which can be described as the following hypothesis testing using a hashing function \( H() \).

- **L0:** Two images \( I \) and \( I' \) are from the same image if the Hamming distance \( D_H(H(I), H(I')) \) is below a threshold \( T \).
- **L1:** Two images \( I \) and \( I' \) are from different images if the Hamming distance \( D_H(H(I), H(I')) \) is above a threshold \( T \).

![Figure 2. Overview of the hash extraction using the polar DFT [7].](image)
3. Proposed weighted hash matching

In image hashing, the real-valued intermediate hash vector is first extracted from an image and then quantized to obtain a binary hash as shown in Figure 1. However; it is almost impossible, or at least not feasible, to include all the discriminant information of an image in a sequence of hash bits. The remaining discriminant information, which is not included in the hash bits, is discarded. To utilize the discarded information further, we compute the probability that the intermediate hash vector belongs to each quantization bin, which is referred to as soft quantization binning. From the soft quantization binning, we compute the probability that the quantized hash in the DB is generated from the query image. The probability value is used for deciding whether the query image corresponds to the quantized hash in the DB.

Let $H_Q$ be the intermediate hash vector of the query image $Q$. Since an image may undergo various distortions during distribution, we regard the query image as a distorted version of the corresponding original image in the DB. Thus the original intermediate hash $H_Q^*$ corresponding to the input query image is assumed to be distributed with the mean $H_Q$ and a distortion bound $\delta_Q$. The distortion bound is the maximum amount of distortion level which is permissible in claiming the identity between $H_Q^*$ and $H_Q$. However; it is intricate to determine the distortion bound. This paper assumes that the distortion bound is proportional to the value of the magnitude of the transform coefficient itself, as is commonly assumed in transform-domain watermark hiding [10-12]. In the case of the polar Fourier-transform hashing, the distortion bound $\delta_Q[j]$ corresponding to the intermediate hash $H_Q[j]$ of the query $Q$ in (1) is given by

$$\delta_Q[j] = \lambda \sum_{i=0}^{L-1} |\beta_{j,i} |F\left(\rho_{j,0} \frac{(2i+1)\pi}{L}\right)| \quad (2)$$

where $\lambda$ is a positive constant that controls the permissible distortion level for the hash matching in determining identity.

![Figure 3. Probability computation based on the triangular-distribution assumption over a scalar quantizer.](image)

Since the original intermediate hash $H_Q^*$ corresponding to the query $Q$ can lie at any point within the distortion bound depending on the distortion types, it is not feasible to estimate an exact probability distribution function (pdf) of $H_Q^*$. We may assign a relevant finite-support pdf $P_{H_Q^*}$ for $H_Q^*$ given $H_Q$ and $\delta_Q$, for example, uniform, triangular, or truncated Gaussian distribution. In this paper, we assume that $H_Q^*[j]$, the $j$-th element of the intermediate hash, follows the triangular distribution given by
\[
P_{H_Q^i;j}(x) = \begin{cases} 
\frac{(x-H_Q[j]+\delta_Q[j])}{\delta_Q[j]^2} & \text{if } H_Q[j] - \delta_Q[j] \leq x \leq H_Q[j] \\
\frac{(H_Q[j]+\delta_Q[j]-x)}{\delta_Q[j]^2} & \text{if } H_Q[j] \leq x \leq H_Q[j] + \delta_Q[j] \\
0 & \text{if } |x - H_Q[j]| > \delta_Q[j]
\end{cases}
\] (3)

In Section 2, the \(K\)-bit gray code (in this paper, \(K = 6\)) is used in extracting binary hash from the intermediate hash. Figure 3 illustrates the triangular distribution of the \(H_Q^j[j]\), where the solid vertical lines represent the quantization boundaries. We propose a probabilistic dissimilarity \(D_W\) between the query’s intermediate hash \(H_Q^j\) (with dimension \(J, J = 64\) in this paper) and the hash \(H_I^j\) in the DB as follows:

\[
D_W(H_Q^j, H_I^j) = 1 - \frac{\sum_{j=0}^{J} \int_{R_Q^i[j]}^{R_I^i[j]} P_{H_Q^i;j}(x) dx}{\sum_{j=0}^{J} \int_{R_Q^i[j]}^{R_I^i[j]} P_{H_Q^i;j}(x) dx}
\] (4)

where \(R_Q^i[j]\) and \(R_I^i[j]\) are the quantization bin of the \(j\)-th dimension corresponding to the \(H_Q^i[j]\) and \(H_I^j[j]\) respectively. We note that we only need the quantization bins of the hash \(H_I\) in the DB, while we utilize the whole probability distribution of \(H_Q(P_{H_Q^i;j})\) of the query image. In practice, the integration in (4) can be computed easily by using one look-up table, parameterized by \((x - H_Q[j] + \delta_Q[j]) / \delta_Q[j]\), of the cumulative distribution function of the triangular distribution in (3).

4. Performance evaluation

To evaluate the proposed method, we tested our method on five thousand images, which include indoor and outdoor scenes, people, vehicles, sporting events, and paintings. The test images are the first 5000 images of the MIR Flickr-25K image dataset. Prior to applying the three chosen hashing methods, we normalized the test images by taking their luminance component and resizing them to \(512 \times 512\). The quantized hashes were extracted from the normalized images using the polar DFT image hashing described in Section 2 and stored in the hash DB. Both the Hamming and the proposed probabilistically weighted distance, \(D_H^j\) and \(D_W\), were computed in hash matching on the hash DB for performance evaluation.

As mentioned in Section 2, the hash matching is formulated as a hypothesis testing in which there are typically two types of error: \(P_{FA}\) and \(P_{FR}\). There is a tradeoff between the two probabilities in selecting the threshold \(T\). For a fair comparison of the proposed distance \(D_W\) with the Hamming distance \(D_H\), which was originally employed for the binary hash, the detection error tradeoff (DET) curve, which plots the \(P_{FR}\) versus the \(P_{FA}\), was used. The DET curve is obtained by measuring both error rates while varying the threshold \(T\) used in the hash matching. \(P_{FA}\) was calculated between all possible pairs of test images in the hash DB. To calculate \(P_{FR}\), we tested the proposed method against the following image processing steps.

- \(\text{WN}\): Adding Gaussian white noise with a mean of zero and a standard deviation of 50.
- \(\text{FT1}\): Applying the 5 by 5 median filter.
- \(\text{FT2}\): Applying the 3 by 3 average filter.
- \(\text{FT3}\): Applying the 3 by 3 unsharp contrast enhancement filter \(H\) given by

\[
H = \begin{bmatrix}
-0.1667 & -0.6667 & -0.1667 \\
-0.6667 & 4.3333 & -0.6667 \\
-0.1667 & -0.6667 & -0.1667
\end{bmatrix}
\]
• FMLR: Applying the frequency mode Laplacian removal (FMLR) [13,14].
• JPG1: JPEG compression of the quality factor 5%.
• JPG2: JPEG compression of the quality factor 75%.

Each test image was subjected sequentially to a set of the selected distortions. We considered four different sets of distortion. By computing the distance between the hashes from the original and the corresponding distorted image using the threshold, $P_{FR}$ was obtained. The resulting DET curve of the polar DFT hashing is shown in Figure 4 for the Hamming distance and the proposed matching method with different values of $\lambda$. The value of $\lambda$ determines the distortion bound that is the amount of distortion allowed by the hashing system, and should be arranged depending on the application scenario. However, we found that with a wide range of the $\lambda$ values the proposed probabilistic dissimilarity $D_w$ performed better than the conventional Hamming distance $D_H$ for all the considered distortions.

![DET curves of the polar Fourier transform hashing for four sets of distortions.](image)

(a) FT1+FT2+FT3+JPG2. (b) JPG1. (c) WN+JPG2. (d) FMLR+JPG2.

5. Conclusion

In this study, we focused on hash matching and proposed a probabilistic dissimilarity based on the distortion bound of a query image to improve hash matching performance. By utilizing the distortion bound of the query image in the hash matching, we compute match a probabilistic score between the query’s intermediate hash and the quantized hash in the DB. Statistical modeling using the distortion bound allows us to compute the probability that the quantized hash in the DB was generated from the query image. In this
study, we applied the proposed method to the DFT-domain image and tested its performance over thousands of images with four different sets of image distortion. For all the considered distortions, the proposed probabilistic dissimilarity was conducive to reducing matching errors as compared with the conventional Hamming distance.

6. Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(2012012876).

References

[1] T. Kalker, J. Haitsma, and J. Oostveen, “Issues with digital watermarking and perceptual hashing,” in Proc. SPIE 4518, Multimedia Systems and Applications IV, 2001.
[2] A. Menezes, P. Oorshot, S. Vanstone, Handbook of Applied Cryptography, CRC press, 1997.
[3] J. Seo, M. Jin, S. Lee, D. Jang, S. Lee, and C. Yoo, “Audio fingerprinting based on normalized spectral subband moments,” IEEE Signal Process. Lett., Vol. 13, No. 4, pp. 209-212, Apr. 2006.
[4] J. Fridrich, “Robust bit extraction from images,” in Proc. IEEE Conf. on Multimedia Computing and Systems, pp. 536-540, 1999.
[5] R. Venkatesan, S. Koon, M.H. Jakubowski, and P. Moulin, “Robust image hashing,” in Proc. IEEE Conf. on Image Processing, pp.664-666, 2000.
[6] J. Seo, J. Haitsma, T. Kalker, C. Yoo, “A robust image fingerprinting system using the Radon transform,” Signal Processing: Image Communication, Vol. 19, pp. 325-339, 2004.
[7] A. Swaminathan, Y. Mao, M. Wu, “Robust and secure image hashing,” IEEE Transactions on Information Forensics and Security, Vol. 1, No. 2, pp. 215-230, June 2006.
[8] F. Lefebvre, B. Macq, and J. Legat, “RASH: Radon soft hash algorithm,” in Proc. European Signal Processing conference, pp. 299-302, 2002.
[9] D. Wu, X. Zhou, and X. Niu, “A novel image hash algorithm resistant to print-scan,” Signal Processing, Vol. 89, No. 12, 2415-2424, Dec. 2009.
[10] R. Wolfgang, C. Podilchuk, and E. Delp, “Perceptual watermarks for digital images and video,” Proc. IEEE, Vol. 87, No.7, pp. 1108-1126, 1999.
[11] M. Barni, F. Bartolini, V. Cappellini, and A. Piva, “A DCT-domain system for robust image watermarking,” Signal Processing, Vol. 66, No. 3, pp. 357-372, 1998.
[12] Q. Cheng and T. Huang, “Robust optimum detection of transform domain multiplicative watermarks,” IEEE Transactions on Signal Processing, Vol. 51, No. 4, pp. 906-924, 2003.
[13] R. Barnett and D. Pearson, “Frequency mode LR attack operator for digitally watermarked images,” Electronics Letters, Vol. 34, No. 19, pp. 1837-1839, 1998.
[14] S. Pereira, S. Voloshynovskiy, M. Madueno, S. Marchand-Maillet, and T. Pun, “Second generation benchmarking and application oriented evaluation,” in Proc. Information Hiding Workshop, pp. 340-353, 2001.