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

From Image Hashing to Video Hashing

Li Weng and Bart Preneel*

Katholieke Universiteit Leuven, IBBT

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

Abstract. Perceptual hashing is a technique for content identification and authentication. In this work, a frame hash based video hash construction framework is proposed. This approach reduces a video hash design to an image hash design, so that the performance of the video hash can be estimated without heavy simulation. Target performance can be achieved by tuning the construction parameters. A frame hash algorithm and two performance metrics are proposed.

1 Introduction

Perceptual hashing is a technique for the identification of multimedia content. It works by computing hash values from robust features of multimedia data. Differing from a conventional cryptographic hash [1], a perceptual hash does not vary, even if the content has undergone some incidental distortion. Therefore, a perceptual hash can be used as a persistent fingerprint of the corresponding content. One can tell whether two multimedia files correspond to the same or similar content(s) by comparing their hash values. Since a hash value is much shorter than the original file in size, this approach is more efficient than direct comparison among multimedia files. A perceptual hash can also be used for authentication, such as message authentication codes (MAC) and digital signatures (DS) [1]. In the former case, the algorithm is designed to support a secret key. The hash value is significantly different when a different key is used, so that only entities knowing the key can generate the correct hash. In the latter scenario, a perceptual hash is electronically signed. Compared to conventional MAC and DS, *perceptual* MAC and DS have the advantage that they do not need to be regenerated when multimedia data undergoes incidental distortion.

* 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 IBBT (Interdisciplinary Institute for BroadBand Technology), a research institute founded by the Flemish Government in 2004, and the involved companies and institutions (Philips, IPGlobalnet, Vitalsys, Landsbond onafhankelijke ziekenfondsen, UZ-Gent). Additional support was provided by the FWO (Fonds Wetenschappelijk Onderzoek) within the project Perceptual Hashing and Semi-fragile Watermarking.

2 Video Hash Construction

Designing a perceptual hash algorithm is challenging. The hash value must be sensitive to significant content modification, but insensitive to content-preserving processing. These traits are defined as *robustness* and *discriminability*. The former is the ability to resist incidental distortion; the latter is the ability to avoid *collisions*, i.e., different contents result in similar hash values. In addition, the hash value must be sensitive to the secret key.

Since there is no strict boundary between similar and dissimilar contents, threshold-based hash comparison is usually used in practice. When comparing a pair of hash values, a decision is made from two hypotheses: 1) $\mathbb{H}_0$ – *they correspond to different contents*; 2) $\mathbb{H}_1$ – *they correspond to similar contents*.

A distance metric is used to measure the similarity between hash values. Only if the distance d is below a threshold t , the contents are judged as similar. The performance can be quantified by the *detection rate* “ p_d ” – probability $\{d < t \mid \mathbb{H}_1\}$, and the *false positive rate* “ p_f ” – probability $\{d < t \mid \mathbb{H}_0\}$. By choosing different values for the threshold, p_d and p_f can be plotted as the receiver operating characteristic (ROC) curve, which characterizes the performance.

Performance evaluation is a difficult task. Deriving the ROC curve generally requires extensive simulation. Typically, a database is used as the ground truth. For each element in the ground truth, several legitimate distortions are applied to produce near-duplicates. The ground truth and the near-duplicates compose an expanded database. Elements in the expanded database are pairwise compared to derive the ROC curve. This procedure is repeated for different keys and different algorithm parameters. Due to excessive computation and the huge storage for test data, the conventional way of performance evaluation may be impractical for video hashing. In order to simplify the task, a framework for video hash construction is proposed in the following.

Since a video is essentially a sequence of frames, a straight-forward way to construct a video hash is to concatenate hash values of video frames. This approach has several advantages: 1) existing image hash algorithms can be used; 2) no need to store an entire video; 3) the video hash performance can be estimated from the image hash performance. In the proposed framework, it is assumed that N frames are extracted from the input video. For each frame, a *frame hash* value $h_{i,i=1,\dots,N}$ is computed by an image hash algorithm $\mathbb{A}$. The video hash $H = \{h_1 \mid \dots \mid h_N\}$ is the concatenation of frame hash values. When two video hash values H_1 and H_2 are compared, *each frame hash h_{i1} of H_1 is compared with the corresponding h_{i2} of H_2 ; the video files are judged as similar if at least T frame pairs are similar*. The ratio $T/N \in (0.5, 1)$ is a constant, denoted by α . The rules are based on the assumption that if two videos are perceptually different, then each pair of extracted frames are also likely to be perceptually different. Assuming $\mathbb{A}$ has performance $\langle p_d, p_f \rangle$ and the N frames are independent, the detection rate and the false positive rate of the overall scheme,

denoted by P_d and P_f , can be formulated as:

$$P_d = \sum_{k=T}^N \binom{N}{k} \cdot p_d^k \cdot (1 - p_d)^{N-k} \quad (1)$$

$$P_f = \sum_{k=T}^N \binom{N}{k} \cdot p_f^k \cdot (1 - p_f)^{N-k}. \quad (2)$$

The above equations show that P_d and P_f increase with p_d and p_f , but decrease with α . For reasonable values of p_d and p_f , i.e., $0 < p_f \ll p_d < 1$, P_d increases with N towards 1, and P_f decreases with N towards 0. The speed of convergence is faster if p_d is closer to 1 or p_f is closer to 0. Therefore, target performance can be achieved by choosing a $< p_d, p_f >$ pair and a suitable N .

Equations (1)-(2) can be adapted to cope with dependent frames. The idea is to divide extracted frames into groups and assume elements in each group depend on each other. Since the dependency varies, only the worst case is considered here as a lower bound. Elements within the same group are assumed to have 100% dependency, i.e., they show the same result for a hypothesis test. Note that dependency does not change the threshold α and the decision rules stay the same. The performance can be estimated by replacing N and T in (1)-(2) with $N' = \lfloor N/a \rfloor$ and $T' = \lceil T/b \rceil$, where a and b are respectively the largest and the smallest cardinalities of the groups. By assuming $a = b = a'$, the dependency actually introduces some expansion in the number of extracted frames N . About a' times more frames are needed for the same performance.

In practice, frames can be missing, and new frames might appear. These factors are treated separately in the following. When two video hash values A and B are compared, it was assumed that each pair of frame hash values A_i and B_i correspond to each other. The complexity is $O(N)$. This is not true if the order of frames has changed. For each A_i , the problem is to find the corresponding $B_{i'}$. A naive approach is to compare each A_i with all frame hash values of B and choose the most similar one. This approach assumes that the frames are completely disordered, so it has the highest complexity $O((N+1)N/2)$. If there is only a time shift, then once a single frame is synchronized, all the rest frames are synchronized too. This only increases the complexity to $O(2N-1)$. A more general case, with complexity $O(lN)$, is that each frame can be synchronized by a maximum of l comparisons. After all, desynchronization increases complexity for hash comparison, but does not affect the formulas.

On the other hand, non-repeatability can be modeled by adding a parameter $\beta \in (0.5, 1]$ to (1)-(2). It is the *overlapping ratio* between the extracted frames from two similar videos. When two hash values of similar videos are compared, a frame hash is either compared to a correct counterpart with probability β , or to a “random” one with probability $1 - \beta$. Therefore, new formulas can be derived by replacing p_d and p_f in (1)-(2) with $p'_d = \beta p_d + (1 - \beta)p_c(t)$ and $p'_f = \beta p_f + (1 - \beta)p_c(t)$, where $p_c(t)$ is defined as the *general collision rate*. For hash length n and threshold t , $p_c(t)$ is the probability that arbitrary two hash

values' Hamming distance is no less than t :

$$p_c(t) = \sum_{k=t}^n \binom{n}{k} \cdot p^k \cdot (1-p)^{n-k}, \quad (3)$$

where p is the probability that two bits coincide, assuming each bit is independent. Since $p_c(t)$ is normally much smaller than p_d and p_f , the effect of non-repeatability is that both P_d and P_f are decreased. Assuming $\alpha = 0.7$, $a' = 10$, $\beta = 0.85$, $n = 144$, $t = 101$, $p = 0.5$, Fig. 1 illustrates the convergence of P_d and P_f by a few examples.

Fig. 1. The convergence of P_d and P_f for practical scenarios.

3 A Frame Hash Algorithm

In this section, a frame hash algorithm is proposed. The algorithm is based on statistics. Since an image is essentially a certain allocation of pixels, the content can be characterized by the statistics of pixel values. In order to reduce collision, statistics of different orders are extracted. The maximum order is limited to four as a compromise between performance and complexity. Specifically, the standard deviation, the third- and the fourth-order auto-cumulants are used in the proposed scheme. The latter two are defined as [2]:

$$C_{3X}(k, l) = E[x(n)x(n+k)x(n+l)], \quad (4)$$

$$\begin{aligned} C_{4X}(k, l, m) = & E[x(n)x(n+k)x(n+l)x(n+m)] \\ & - C_{2X}(k)C_{2X}(l-m) - C_{2X}(l)C_{2X}(k-m) \\ & - C_{2X}(m)C_{2X}(k-l), \end{aligned} \quad (5)$$

where X is a zero-mean vector, E means expectation, and $C_{2X}(k) = E[x(n)x(n+k)]$. A reason to use higher-order cumulants is that they are immune to Gaussian noise [2]. Additionally, it is known that natural images are non-Gaussian [3]. The assumption that content-preserving processing preserves non-Gaussianity has been validated by applying the fourth-order cumulant in image hashing [4].

The algorithm starts with pre-processing. An extracted frame is first converted to gray and resized to a canonical size 512×384 . The output is filtered by an averaging filter and a median filter. Histogram equalization is then performed. These steps limit the computation complexity, remove slight noise, and stabilize the content. The pre-processed image is divided into 256×192 pixel blocks with 50% overlapping. Pixels in each block are scanned into a vector and normalized to be zero-mean. From each vector, the standard deviation, the third- and the fourth-order cumulants are computed. Each statistic is computed twice to increase robustness. The first time is for the original block, the second time is for a transposed version of the original block. The mean value is the output. In order to make the hash compact, only one value is kept for each statistic. Each block is allocated with 16 bits for quantization. Specifically, 3 bits for the standard deviation, 6 bits for the third-order cumulant, and 7 bits for the fourth-order cumulant. Higher-order statistics are given finer quantization, because they are more representative. There are 9 blocks, so a hash value has 144 bits. There are two steps of randomization. The first is key-based feature extraction. The parameters k , l , and m in (4)-(5) are decided by a secure pseudo-random number generator (PRNG), which accepts a secret key. The second step is dithering. After quantization, the binary output is XORed with a dither sequence generated by the secure PRNG. The distance metric for hash value comparison is the bit error rate (BER), aka the normalized Hamming distance.

The performance of the algorithm has been evaluated by simulation. In the simulation, 432 different natural scene images in the JPEG format are used. They consist of several categories: architecture, landscape, sculpture, objects, humanoid, and vehicle. Each category includes 72 images of three canonical sizes: 1600×1200 , 1024×768 , and 640×480 . There are 10 operations defined to generate distorted but authentic images, each with 5 levels, listed in Table 1.

Table 1. The robustness test.

Distortion name	Parameter range, step	Average bit error rates					
1. Rotation	Angle: $2^\circ - 10^\circ$, 2°	0.086	0.142	0.174	0.194	0.210	
2. Gaussian noise	Standard deviation: 10 – 50, 10	0.020	0.026	0.032	0.041	0.051	
3. Central cropping	Percentage: 2% – 10%, 2%	0.078	0.127	0.159	0.185	0.209	
4. JPEG compression	Quality factor: 50 – 10, 10	0.001	0.014	0.010	0.016	0.033	
5. Scaling	Ratio: 0.5 – 0.1, 0.1	0.003	0.008	0.012	0.022	0.055	
6. Median filter	Window size: 3 – 11, 2	0.005	0.012	0.021	0.030	0.036	
7. Gaussian filter	Window size: 3 – 11, 2	0.003	0.003	0.003	0.003	0.003	
8. Sharpening	Strength: 0.5 – 0.1, 0.1 ¹	0.022	0.022	0.023	0.023	0.023	
9. Salt & pepper	Noise density: 0.01 – 0.05, 0.01 ¹	0.009	0.015	0.021	0.026	0.030	
10. Row/col. removal	No. of removals: 5 – 25, 5	0.009	0.012	0.015	0.017	0.020	

In the robustness test, a distorted image is compared with the original by the hash values. For each comparison, a key is randomly chosen from 500 ones.

¹ These are parameters for Matlab functions `fspecial()` and `imnoise()`.

The average BERs are listed in Table 1. The maximum BER is about 0.2, given by rotation and cropping. In the discrimination test, there are 22032 images in total. Any two of them are compared by their hash values. A default key is used. The average BER between different images is 0.33 with standard deviation 0.045. Therefore, the BER margin against incidental distortion is about 0.1. The ROC curve is plotted in Fig. 2. A state-of-the-art image hash algorithm [5] is tested by the same procedure for performance comparison. Its ROC curve for a default key is also plotted in Fig. 2. The figure shows that the proposed scheme has superior performance.

Fig. 2. ROC curves.

4 New Performance Metrics

Although the ROC curve tells the performance, sometimes it is useful to measure robustness and discriminability separately. In the following, two metrics are proposed for this purpose, defined as the *robustness index* and the *discrimination index*. They are suitable for hash comparison schemes based on the Hamming distance. Before the details, note that the robustness can be represented by the $p_d(t)$ curve, which is the detection rate p_d versus the threshold t . Similarly, the discriminability is represented by the $p_f(t)$ curve.

The discrimination index is basically the difference between the $p_f(t)$ and the $p_c(t)$ curves. The idea is to use the $p_c(t)$ curve as a reference, because it is close to the theoretical (best) $p_f(t)$ curve. Although the general collision rate is larger than the theoretical false positive rate according to the definition, practical schemes hardly achieve it, i.e., p_f is almost always larger than p_c . Therefore, the general collision rate can be used as a reference or even a lower bound of the false positive rate in practice. The $p_c(t)$ curve is derived by assuming $p = 0.5$ in (3) for different values of t . The $p_f(t)$ curve comes directly from the ROC curve. The KL-divergence is used to measure the distance between the two. The discrimination index is defined as

$$D(p_c||p_f) = \sum_t p'_c(t) \frac{p'_c(t)}{p'_f(t)}, \quad (6)$$

where $p'_c(t) = p_c(t) / \sum p_c(t)$ and $p'_f(t) = p_f(t) / \sum p_c(t)$ are normalized versions of $p_c(t)$ and $p_f(t)$. A larger index means better discrimination. The indices of the proposed scheme and the reference scheme are -2.09 and -2.11 respectively. Therefore the proposed scheme has better discrimination. This is consistent with the ROC curve comparison.

Since there is no non-trivial upper bound for p_d , the robustness index is defined differently. The idea is to model $p_d(t)$ by (3). Note that when hash values corresponding to similar contents are compared, p in (3) should be close to 1. Therefore, p can be interpreted as the average detection rate per bit. The robustness index is defined as the p value which results in a $p_c(t)$ curve that is closest to the $p_d(t)$ curve by (6). A larger index means stronger robustness. The indices of the proposed scheme and the reference scheme are 0.96 and 0.99 respectively. Interestingly, the reference scheme shows stronger robustness. It is confirmed by a robustness test. The ROC curves and the indices imply that the superior overall performance of the proposed scheme is due to a good balance between robustness and discriminability.

5 Conclusion

In this work, it is proposed that a perceptual video hash can be constructed from the perceptual hash values of video frames. This framework has the advantage that the performance can be estimated from the performance of the frame hash. A frame hash algorithm based on statistics is proposed. Simulation shows that it outperforms an existing algorithm. As a complement to the ROC curve, two metrics are proposed to measure robustness and discriminability.

References

1. Schneier, B.: Applied Cryptography: Protocols, Algorithms, and Source Code in C. 2nd edn. John Wiley & Sons (1996)
2. Mendel, J.: Tutorial on higher-order statistics (spectra) in signal processing and system theory: theoretical results and some applications. In: Proc. of the IEEE. (1991)
3. Krieger, G., Zetzsche, C., Barth, E.: Higher-order statistics of natural images and their exploitation by operators selective to intrinsic dimensionality. In: Proc. of the IEEE Signal Processing Workshop on Higher-Order Statistics. (1997) 147–151
4. Weng, L., Preneel, B.: On secure image hashing by higher-order statistics. In: Proc. of IEEE International Conference on Signal Processing and Communications. (2007) 1063–1066
5. Swaminathan, A., Mao, Y., Wu, M.: Robust and secure image hashing. IEEE Transactions on Information Forensics and Security **1**(2) (June 2006) 215–230