A FRAMEWORK FOR SOFT HASHING AND ITS APPLICATION TO ROBUST IMAGE HASHING

E. McCarthy, F. Balado, G.C.M. Silvestre and N.J. Hurley

University College Dublin, Belfield, Dublin 4 – Ireland

ABSTRACT

An increasing interest in the soft hashing problem has been witnessed in recent times. Techniques implementing soft hashing intend to mirror the behaviour of cryptographic hashing when the information to be hashed can be subject to different kinds of distortions. Many heuristic techniques for undertaking soft hashing of images and other multimedia data have been devised. Except for some attempts, a framework giving solid guidelines to solve the problem is largely lacking. In this paper we provide one possible approach to undertake the modelling of robust soft hashing, detailing the basic problems involved. We show how some prior schemes partly fit inside our model.

1. INTRODUCTION

Soft hashing, also known as robust hashing or perceptual hashing, consists of summarising multimedia data, so as to obtain a concise representation called a hash value (also, fingerprint, message digest, or label). The hash generation procedure should be such that perceptually similar data yield the same hash value. The soft hashing problem is interesting for a number of scenarios which in most cases involve indexing of multimedia databases and/or authentication, where the hash provides a compact representation which can be used to identify the data efficiently.

Different hashing applications impose different requirements. Usually soft hashing should significantly reduce the dimensionality of the data, without substantially increasing the probability of collision (or false positive rate), i.e. the probability of having the same hash value for any two perceptually different data objects. In the case of authentication applications, the dimensionality reduction should be made using a one-way key-dependent function. In indexing applications, robustness to distortions which do not affect the perceptual similarity of the multimedia data is essential.

Up to now, many different robust hashing schemes have been proposed for multimedia hashing (see for instance for the case of images [1, 2, 3, 4]). Nevertheless, a more general approach allowing the problem to be addressed in a systematic way is largely lacking. Previous proposals by Johnson and Ramchandran [5] and by Mıhçak and Venkatesan [6] have partially tried to fill this gap already.

In this paper we propose a soft-hashing framework to gain insight into the main design lines of this type of system. We emphasise its robustness aspect, as required by the application to database indexing. We identify the central blocks of the problem, which, although already hinted at by different researchers in one way or another, are presented here in a unified way. Last, we propose an application of the developed methodology to the problem of image hashing.

Enterprise Ireland is kindly acknowledged for supporting this work under the project ATRP 2002/230 .

2. SOFT HASHING FOR DATABASE INDEXING

In the following, we will consider a soft hashing system for database indexing. Due to this, key-related security issues will not be discussed, and we will focus our attention on the design of distortion-resistant soft hashing methods. The multimedia signal to be hashed will be denoted without loss of generality by a continuous-valued n-dimensional vector $\mathbf{x} = (x_1, \ldots, x_n)$. In the general case, this vector may undergo some possibly random distortion function that we can write as $f(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$. Our objective is to map the signal $\mathbf{x}$ to an index belonging to a finite set $\mathcal{H}$, such that the index is as independent as possible of the distortion function applied. A working hypothesis is that distortions have to be constrained so that the distorted signal $\tilde{\mathbf{x}} = f(\mathbf{x})$ is not too different from $\mathbf{x}$ under some perceptually meaningful criterion (see Sect. 3.1).

As depicted in Fig. 1, it is possible to divide database indexing systems using soft hashing into three quite independent blocks, namely:

- Synchronisation. As in communications problems it is necessary that the signal $\mathbf{y} = (y_1, \ldots, y_m)$ presented to the hashing function always matches the same indices, in spite of possible desynchronisations undergone by $\mathbf{x}$. Distortions that affect synchronism can be of very different nature, such as warpings, croppings, rotations, among others. It is not possible to insert synchronisation pilots in $\mathbf{x}$, as is common practice in communications systems. Most existing soft hashing systems try to solve this issue through the use of feature mappings. These mappings are just functions $s(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$, with $m \leq n$, that exploit geometrical invariances of $\mathbf{x}$ and that are therefore dependent on the nature of the multimedia signal. Examples of such mappings for images may be moments of different orders, or the Fourier-Mellin transform [7].
- Hashing. Once the synchronisation of $\mathbf{y}$ is assured we can safely design a hashing function

$$h(\cdot) : \mathbb{R}^m \to \mathcal{H}. \tag{1}$$

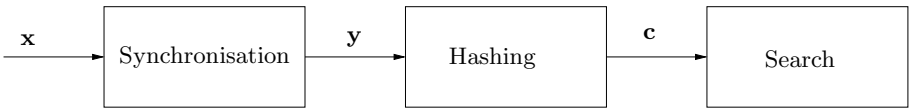

Fig. 1. A Model for a Database Indexing System using Soft Hashing

A mapping from an m-dimensional continuous vector to an index belonging to a finite set $\mathcal{H}$ can be seen as quantization [5] or clustering. It is clear that only the knowledge of the statistics of $\mathbf{y}$ can lead to an optimal design of this multidimensional quantizer. Nevertheless, quantization is not the only ingredient of the problem, as, even if the vector at the input of $h(\cdot)$ is perfectly synchronised, the *amplitudes* of its samples could be modified by the distortion function $f(\cdot)$, having a vector $\tilde{\mathbf{y}} = \mathbf{y} + \mathbf{n}$ at the input of the hashing function. In this situation, codes may be required to recover from the errors caused by this distortion. This block constitutes the main subject of this paper and is discussed in more detail in the following sections.

- Search. Searching takes place when we need to compare a hash value with those precomputed and stored in a multimedia database. This search can be very expensive for huge databases, and for this reason we would like our hash size to be the shortest possible. In any case, the minimum size can be quite large, depending on the nature of the hashed signals, and smart strategies such as Viterbi search might be suitable in order to deal with complexity issues.

Notice that two dimensionality reductions are involved: one potentially due to the synchronisation block, and the other one due to the hashing function itself. One would like the dimensionality reduction due to synchronisation to be the less important one, so that more degrees of freedom are possible for the design of the hashing function itself.

3. MODELLING THE HASHING BLOCK

In [5] guidelines are given for designing a perfectly *secure* hashing block, whose hash values can only be computed by the owner of a secret key. The hashing block is proposed to be source coding with side information at the decoder, where this side information is a key-dependent dither which is added to $\mathbf{y}$ before source coding. Alternatively, we focus here on the robustness of the hash values to further distortions that affect the original multimedia signal. Still, we will see that source and channel codes are also involved. Notice that we cannot rely here on a dither key-dependent vector for each database signal as we cannot rely on the a priori knowledge of any concrete signal.[1]

We will assume in the following perfect synchronisation at the input of the hashing block, such that the indices of $\mathbf{y}$ and its amplitude-distorted version $\tilde{\mathbf{y}}$ match. Also, we assume that the samples y_i, $i = 1, \ldots, m$, are i.i.d. and that they follow a certain distribution $f_Y(y)$ with variance σ_y^2. Fig. 2 shows the main building blocks of our proposal.

[1]Using the same key for every signal would break the security assumptions in [5].

3.1. Quantization

The first objective is to design the quantizer (1) that maps $\mathbf{y}$ to an integer index $k \in \mathcal{H} = \{1, \ldots, 2^{mR}\}$. This index k may be represented using mR bits (assumed integer). Hence, the dimensionality reduction is given by the rate R. The quantizer should reach a compromise between the two following conflicting properties:

1. Two "similar" realisations of the stochastic process $\mathbf{Y}$ should map to the same index k.
2. $|\mathcal{H}| = 2^{mR}$ must be large enough to distinguish a "sufficient" number of instances of $\mathbf{Y}$, i.e., to decrease the probability of collision.

Both properties require a distortion criterion for their objective evaluation. Note that we are proposing to measure distortion before the hashing block, and not afterwards as in [6]. Although the latter option is also coherent, we will see below that our definition has several advantages. We may define an average distortion criterion using the expectation of a sample-by-sample error function between any two vectors $\mathbf{y}$ and $\hat{\mathbf{y}}$, such as the square error:

$$D \triangleq E\{d(\mathbf{y}, \hat{\mathbf{y}})\} = E\left\{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2\right\}. \qquad (2)$$

Other more accurate criteria using perceptual models are also possible, leading to different results.

Let us assume first that $\tilde{\mathbf{y}} = \mathbf{y}$ after the synchronisation stage; we will deal with the general case in Sect. 3.2. In this case, the problem would come down to that of source encoding with a given fidelity criterion. This means that $\mathbf{y}$ may be reconstructed from $k = h(\mathbf{y}) \in \mathcal{H}$ using a certain function $g(\cdot)$:

$$\hat{\mathbf{y}} = g(k) = g(h(\mathbf{y})), \qquad (3)$$

under the constraint $D \leq D_{\max}$. The limits of the optimal solution to this problem are given by Rate-Distortion (R-D) theory, which determines the minimum rate $R_{\min}$ of the quantizer (1) for the distortion constraint to hold.

In our problem we may reinterpret $\sqrt{D_{\max}}$ as the average radius of the ball in $\mathbb{R}^m$ that maps to the same hash value, i.e., as the *granularity* or resolution of the soft hashing function. The reconstruction function (3) would just give the centroids or representatives of those regions. In this case, R-D theory establishes the theoretically minimum hash size for achieving the resolution target $D_{\max}$ (i.e., the reconstruction distortion). Interestingly, a quantizer built along these guidelines guarantees the property of *uniformity* of the hash values (quantization indices) over the distribution of the multimedia signal to be hashed.

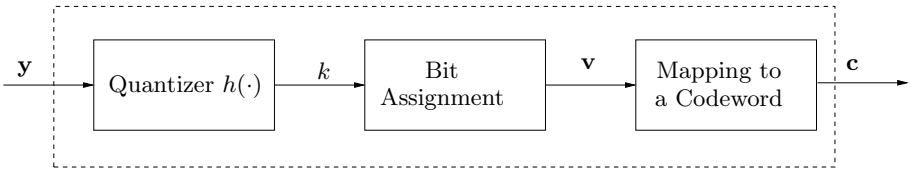

Fig. 2. A Model of the Hashing Block

Moreover, once the quantizer is built we may compute beforehand the probability of collision between any two similar signals (P_c) using the statistical characterisation of $\mathbf{y}$:

$$P_c = \int f_{\mathbf{Y}}(\mathbf{y}) \int_{\mathcal{I}(\mathbf{y})} f_{\mathbf{Y}}(\mathbf{y}') \, \mathrm{d}\mathbf{y}' \mathrm{d}\mathbf{y}, \tag{4}$$

with $\mathcal{I}(\mathbf{y}) \triangleq V_{h(\mathbf{y})} \cap S(\mathbf{y})$, that is, the intersection of the Voronoi (quantization) region corresponding to $k = h(\mathbf{y})$ with a ball of similar vectors $S(\mathbf{y})$ (measured using (2)), centered at $\mathbf{y}$.

3.2. Bit Assignment and Channel Code

In many practical cases we may not assume that $\tilde{\mathbf{y}} = \mathbf{y}$ as done in the preceding section, due to intentional or unintentional distortions such as lossy compression and others. Once the quantization function (1) is chosen, notice that a vector lying near the border of a Voronoi region could be easily moved to a neighbouring region (that maps to another different index) with just a small distortion.

Consequently, let us assume next that, after the synchronisation stage, the input to the hash function is distorted by a zero-mean additive noise $\mathbf{n}$, independent of $\mathbf{y}$ and with covariance matrix $\Gamma_n = \sigma_n^2 I$, i.e. $\tilde{\mathbf{y}} = \mathbf{y} + \mathbf{n}$.

In order to cope with this situation several strategies are possible. One possible approach would involve taking into account the distortion $\mathbf{n}$ in the codebook design. In the literature, this is known as noisy source coding [8], whose usual design target is a distortion constraint between the noiseless signal $\mathbf{y}$ and the reconstruction of the quantized noisy signal, $g(h(\tilde{\mathbf{y}}))$.

Notice that, under noise distortions, the optimal quantizer in Sect. 3.1 may no longer be useful. This is so because its optimality in the R-D sense implies that the more likely values are more finely quantized, and then, due to the same reason, these more probable values are less robust to noise. We will give an example of this behaviour and discuss a suitable quantizer in Sect. 5.

We may also use a binary channel code $\mathcal{C}$ at the quantizer output, in order to fight the errors induced by noise. A similar strategy was followed in [6]. In general, for using a code we will need to convert the quantization index k to a binary vector representation $\mathbf{v}$ (bit assignment). This vector has to be mapped to the nearest codeword vector $\mathbf{c} \in \mathcal{C}$, and so the length of both vectors is assumed to be equal.

To illustrate the importance of the bit assignment issue consider a one-dimensional codebook having 16 uniformly placed centroids. If we use a natural bit assignment we can have neighbouring regions which differ in as much as four bits, such as 1000 and 0111. If we use instead a Gray code then the number of bit differences between neighbouring regions will never be greater than one, allowing for better performance of the code. This assignment is not so obvious in higher dimensions. Several methods have been proposed for near-optimal bit assignment in the case where a multidimensional codebook is used. These include iterative optimisation methods such as simulated annealing [9] and the Binary Switching Algorithm [10].

A code rate R_c implies that the number of coded hash values is 2^{mR_cR}, instead of 2^{mR}. Then, with a code we are actually trading resolution power for robustness of the hash system. We may denote the whole hashing block in Fig. 2 as $h_c(\cdot) : \mathbb{R}^m \to \mathcal{C}$. The robustness of the scheme can be determined using the probability of error

$$P_e = P\{h_c(\tilde{\mathbf{y}}) \neq h_c(\mathbf{y})\}, \tag{5}$$

or the corresponding bit error rate (BER). This amount is a function of a given granularity-to-distortion ratio, that may be defined as

$$\mathrm{GDR} \triangleq 10 \log_{10} \frac{D_{\max}}{\sigma_n^2}. \tag{6}$$

4. APPLICATION TO IMAGE HASHING

We describe next some guidelines on how this framework would be applied to image hashing. For rotation, scale and translation invariance we may use the Fourier-Mellin transform as a synchronisation block, as in [7], for instance. Afterwards, the discrete cosine transform (DCT) may be applied, as its low-medium frequency coefficients are known to be well modelled by the generalised Gaussian distribution. Other similar strategies yielding modelable inputs to the hashing block are also possible. As the input $\mathbf{y}$ to the quantization block can now be modelled statistically, the design of an optimal quantizer is possible. An approximation to it can be obtained with the generalised Lloyd algorithm using a suitable initialisation. As discussed in the previous section, we have to take into account the potential weaknesses of an optimal codebook if further distortions are present in our system.

In the latter case, the hashing block requires the application of a suitable bit assignment method as described in Sect. 3.2. The resulting bitstrings may be concatenated in a systematic or possibly key-dependent pseudorandom manner, to give the resulting vector $\mathbf{v}$. The choice of an appropriate code for the final step is discussed in Sect. 5.

5. EXPERIMENTAL RESULTS

In order to illustrate the proposal above we assume a scenario where the image to be hashed is represented by a zero-mean i.i.d. Laplacian vector $\mathbf{y}$ with $m = 1000$ samples. We

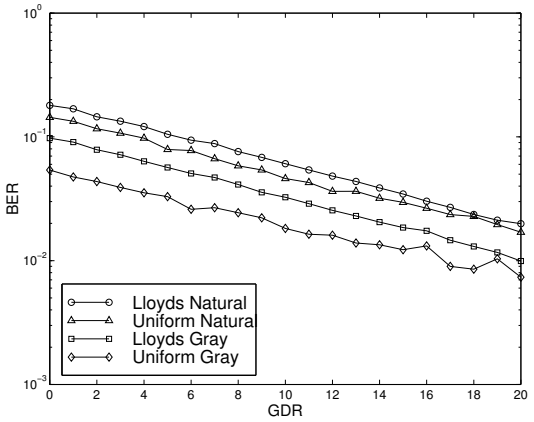

Fig. 3. Performance of Lloyd and uniform quantizers without using a code, and using natural and Gray bit assignments.

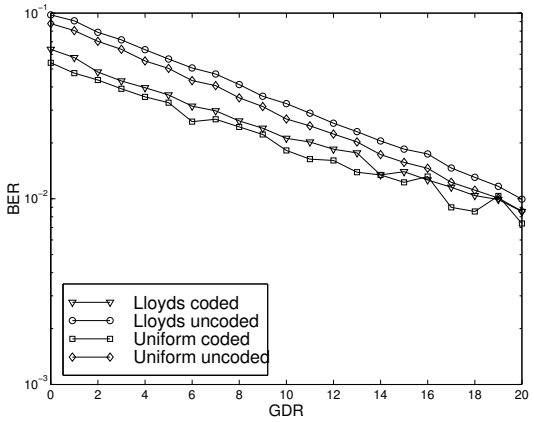

Fig. 4. Hashing block performance for Lloyd and uniform quantizers, using Gray bit assignments and a R-S (255,55) code.

apply unidimensional quantizers separately to each dimension and test both natural and Gray bit assignment. All quantizers are built with 8 centroids. Fig. 3 shows the performances of a quantizer designed with the Lloyd method and the optimal uniform quantizer under i.i.d. Gaussian distortion with variance dependent on the GDR. Notice that the practical values of this parameter are high if we want to limit the perceptual impact of the added noise, when using a coarse hash granularity. We observe that the uniform quantizer shows better performance, due to the reasons explained in Sect. 3.2. In addition, the importance of the bit assignment is seen.

At the final step, the use of a convolutional seemed a suitable choice for easily providing arbitrarily long codewords. However, preliminary tests resulted in quite high BER's, possibly due to the codewords being clustered in this type of code. As decoding (i.e., mapping of $\mathbf{v}$ to a codeword) is not preceded in this problem by encoding, clustered codewords may affect negatively the decoding process. This suggests that the most suitable codes in this scenario should have uniformly distributed codewords. In this sense Reed-Solomon codes proved better, but other codes may largely improve the performance results obtained with these codes and shown in Fig. 4. Notice that the relatively high probabilities of error will be much lower in a higher dimensional case.

In conclusion, in this paper we have proposed a framework giving guidelines to build robust soft hashing methods. A proposal on the application of the methodology to image hashing has been given, together with empirical tests for validation.

6. REFERENCES

[1] R. Venkatesan, S.M. Koon, M.H. Jakubowski, and P. Moulin, "Robust image hashing," in *Procs. of the IEEE International Conference on Image Processing*, Vancouver, Canada, 2000.

[2] M. Mıhçak and R. Venkatesan, "New iterative geometric methods for robust perceptual image hashing," in *Procs. of ACM Workshop on Security and Privacy in Digital Rights Management*, Philadelphia, USA, 2001.

[3] C. Kailasanathan, R. Safavi Naini, and P. Ogunbona, "Compression tolerant DCT based image hash," in *Procs. of the 23rd Intnl. Conf. on Distributed Computing Systems Workshops*, Rhode Island, USA, May 2003, pp. 562–567.

[4] F. Lefebvre, J. Czyz, and B. Macq, "A robust soft hash algorithm for digital image signature," in *Procs. of the IEEE International Conf. on Image Processing*, Barcelona, Spain, September 2003, vol. 2, pp. 495–498.

[5] M. Johnson and K. Ramchandran, "Dither-based secure image hashing using distributed coding," in *Procs. of the IEEE International Conf. on Image Processing*, Barcelona, Spain, September 2003, vol. 2, pp. 751–754.

[6] M. Mıhçak and R. Venkatesan, "A perceptual audio hashing algorithm: A tool for robust audio identification and information hiding," in *Procs. of the 4th Information Hiding Workshop*, Pittsburgh, USA, 2001.

[7] J. Fridrich, "Visual hash for oblivious watermarking," in *Procs. of SPIE: Security and Watermarking of Multimedia Contents*, San José, USA, 2000.

[8] E. Ayanoglu, "Optimal quantization of noisy sources," in *Procs. of ICASSP*, Seattle, USA, April 1988, vol. 1, pp. 569–572.

[9] N. Farvardin, "A study of vector quantization for noisy channels," *IEEE Trans. on Information Theory*, vol. 36, no. 4, pp. 799–809, July 1990.

[10] K. Zeger and A. Gersho, "Pseudo Gray coding," *IEEE Trans. on Communications*, vol. 38, no. 12, pp. 2147–2158, December 1990.