Visual Hash For Oblivious Watermarking

Jiri Fridrich

Center for Intelligent Systems, SUNY Binghamton, Binghamton, NY 13902-6000

Mission Research Corporation, 1720 Randolph Rd SE, Albuquerque, NM 87501

ABSTRACT

Digital watermarks have recently been proposed for authentication and fingerprinting of both video data and still images and for integrity verification of visual multimedia. In such applications, the watermark must be oblivious and has to depend on a secret key and on the original image. It is important that the dependence on the key be sensitive, while the dependence on the image be continuous (robust). Both requirements can be satisfied using special image digest (hash) functions that return the same bit-string for a whole class of images derived from an original image using common processing operations including rotation and scaling. It is further required that two completely different images produce completely different bit-strings. In this paper, we extend our previous work on robust image digest functions describing ideas how to make the hash function independent of image orientation and size. The robust image digest can be clearl utilized for other applications, such as a search index for an efficient image database search.

Keywords: Digital watermarking, authentication of video and images, digital image processing, search index, image database, visual hash functions

1. INTRODUCTION

Digital watermark is a perceptually invisible pattern embedded in a digital image. The watermark can carry information about the owner of the image or the recipient (watermarking for copyright protection, fingerprinting, or traitor tracing), the image itself (watermarking for tamper detection and authentication), or some additional information accompanying the image (image caption embedding). Watermarking schemes can be divided into two groups depending on whether or not the original image is required for watermark extraction. In non-oblivious watermarking, the original image is needed for watermark extraction. Although this makes non-oblivious techniques more robust to attacks, the necessity of having the original image is clearly a disadvantage that severely limits the applicability of non-oblivious techniques. In oblivious techniques, the watermark can be extracted from the watermarked / attacked image without access to the original image. In some watermarking techniques, one must have access at least to a hash of the image (or a hash of the whole video) in order to recreate the watermark sequence at the receiving end in order to be able to correlate the watermark with the watermark extracted from the image itself[1]. Such techniques are not truly oblivious because the hash needs to be exchanged prior to watermark detection.

Secure oblivious watermarking of videos for fingerprinting or authentication requires watermarks that depend on each frame. Indeed, one watermark pattern inserted into each frame would lead to a very vulnerable watermarking scheme with a serious security gap. It has been shown that by processing the images (frames), it is possible to statistically recove a good approximation to the watermark pattern[2]. However, the requirement of the technique to be oblivious means that either the watermark depends on the frame index or it is determined by the frame itself. Obviously, the latter case leads to more versatile schemes. A reliable method for generating a good approximation of the watermark from the image itself (even after watermarking and attacks) will clearly lead to more useful and elegant oblivious watermarking schemes.

Tewfik et al.[1] describe a watermarking technique in which a user-defined noise-like signature is modulated with a perceptual mask calculated from small blocks using perceptual masking. The same signature is used for all video-frames. The watermark pattern in this application is frame dependent and does not depend on the frame index. However, the frame dependency is not too strong because the perceptual mask can be calculated from each frame, which makes the technique equivalent to watermarking with a fixed watermark pattern.

Image watermarking for tamper detection leads to a similar situation as watermarking videos. Each digital image with a digital camera or digital video-camera would be watermarked on the fly so that later we can prove image integrity o indicate blocks in the image that have been tampered with. For a comprehensive review of watermarking techniques fo tamper detection and common security problems see the paper by Fridrich[3]. Again, in this particular application, using one pattern that does not depend on the image would be insecure because analyzing a relatively small number of images may reveal the watermark pattern[2].

What is needed in both applications discussed above is a watermark W that depends sensitively on a secret ke K and continuously on the image I:

1. $W(K, I)$ is uncorrelated with $W(K, I')$ whenever images I and I' are dissimilar;
2. $W(K, I)$ is strongly correlated with $W(K, I')$ whenever I and I' are similar (I' is the image I after an attack comprising of a rotation, scale, and grayscale modifications);
3. $W(K, I)$ is uncorrelated with $W(K', I)$ for $K \neq K'$.

Linnartz and Cox [4,5] proposed similar requirements for watermarking digital video disks (DVD) for copy control. The requirements 1–3 could be satisfied provided we have a robust image digest function H that would return the same N bits (or almost the same N bits) for all images I that underwent a combination of a rotation R_φ by an angle φ, scaling S_α by a factor α, and typical grayscale operations G. Noise adding, filtering, lossy JPEG compression, gamma correction, and histogram equalization are examples of typical grayscale operations. So, if the robust hash function H depends on a parameter K (secret key), we require that

$$H_K(R_\varphi \circ S_\alpha \circ G(I)) = const. \in \{0,1\}^N, \text{ for all } \varphi, \alpha, \text{ and } G. \quad (1)$$

In our previous work[6], we have introduced a robust bit extraction method that satisfied Eq. 1 for a wide range of grayscale operations G but which did not include rotation and scale. In Section 2, we describe this method, and in Section 3, we present new ideas how to extent the method in order to satisfy Eq. 1 including the invariance to rotation R_φ and scaling S_α. In Section 4, we explain how to use the calculated visual hash to synthesize a Gaussian sequence that gradually loses the correlation with the original sequence obtained from all correct hash bits. Section 5 contains the conclusion and directions for future research.

2. VISUAL HASH FUNCTION FOR IMAGES

In this section, we describe a mechanism for robust extraction of bits from images so that all similarly looking blocks, whether they are watermarked, unwatermarked or attacked by gray scale modifications, will produce almost the same bit sequence of a specified length N. The method is based on the observation that if a low-frequency DCT coefficient of an image is small in absolute value, it cannot be made large without causing visible changes to the image. Similarly, if the absolute value of a low-frequency coefficient is large, we cannot change it to a small value without influencing the image significantly. To make the procedure dependent on a key, the DCT modes are replaced with low frequency, DC-free, (i.e., having zero mean) random smooth patterns generated from a secret key (with DCT coefficients equivalent to projections onto the patterns). For each image, a threshold Th is calculated so that on average 50% of projections have absolute value larger than Th and 50% are in absolute value less than Th. This maximizes the information content of the extracted N bits and increases the robustness of the visual hash function.

Using a secret ke K (a number uniquely associated with an author, movie distributor, or a digital camera) we generate N random matrices with entries uniformly distributed in the interval [0, 1]. Then, a low-pass filter is repeatedly applied to each random matrix to obtain N random smooth patterns P_i, $1 \leq i \leq N$. An example of four random patterns and thei smoothened versions are shown in Figure 1. All patterns are then made DC-free by subtracting the mean from each pattern. Considering the block and the pattern as vectors, the image I is projected on each patter P_i, $1 \leq i \leq N$, and its absolute value is compared with the threshold Th to obtain N bits b_i:

$$\text{if } |B \cdot P_i| < Th \qquad b_i = 0,$$
$$\text{if } |B \cdot P_i| \geq Th \qquad b_i = 1.$$

Since the patterns P_i have a zero mean, the projections do not depend on the mean gray value of the block and only depend on the variations within the block itself. The distribution of the projections is image dependent and should be adjusted accordingly so that approximately half of the bits b_i are zeros and half are ones. This will guarantee the highest information content of the extracted N-tuple. This adaptive choice of the threshold becomes important for those image operations that significantly change the distribution of projections, such as contrast adjustment or gamma correction.

The robustness of this visual hash function has been tested on real imagery with very encouraging results (see Table 1). The visual hash algorithm can reliably extract over 48 correct bits (out of 50 bits) from a small 64×64 image for the following image processing operations: 15% quality JPEG compression (as in PaintShop Pro 4.12), additive uniform noise with amplitude of 30 gray levels, ±50% contrast adjustment, ±25% brightness adjustment, dithering to 8 colors, multiple applications of sharpening, blurring, median, and mosaic filtering, histogram equalization and stretching, edge enhancement, and gamma correction in the range 0.7–1.5. Taking the negative of the image returns all 50 correct bits as expected. Quite understandably, operations like embossing change the visual hash significantly. This could be remedied by calculating the visual hash from robustly extracted edges rather than the whole image. Geometrical modifications, such as rotation, shift, and change of scale, also produce vastly different visual hash bits.

Image operation	**Random smooth patterns**	**Random patterns (non-smooth)**
85, 50, 25, 15, 5% quality JPEG	49.69, 49.81, 49.63, 49.00, 47.69	47.88, 46.19, 44.63, 44.00, 41.82
Adding uniform noise in [–7, 7], [–22, 22], [–36, 36]	49.56, 48.81, 47.69	45.50, 37.13, 32.06
–50, –25,+25, +50% contrast	49.88, 49.94, 49.94, 48.75	49.50, 49.69, 49.94, 46.50
–30, –15,+15, +30% brightness	47.31, 49.94, 49.81, 47.44	45.56, 49.81, 49.81, 46.06
Posterizing to 2, 4, 8, 16 colors	39.56, 45.50, 48.25, 49.19	34.00, 40.00, 44.44, 47.56
Sharpen 1×, 2×	49.81, 49.56	46.56, 41.56
Histogram equalization	48.06	46.25
Histogram stretching	49.88	49.88
Blurring 1×, 4×	49.50, 48.69	43.13, 40.19
0.7, 1.5 gamma correction	48.50, 48.75	47.19, 48.25
Median filter 3×3	49.56	44.94
Mosaic filter 6×6	48.50	40.19
Edge enhancement	48.75	35.94
Erode	44.31	38.13
StirMark 1.0 1×, 2×, 3×	45.31, 39.75, 35.13	35.94, 31.63, 29.13
Emboss	26.19	27.13

Table 1 Average number of correct visual hash bits out of 50 bits after some image processing operations.

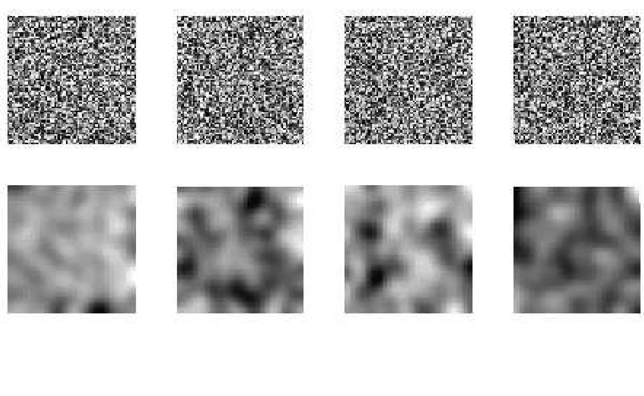

Figure 1 Examples of four random patterns and their smoothened version

The number of correct hash bits should be compared to the number of correct hash bits due to pure chance given some priori probability distribution of 1's and 0's in the bit sequence (see Figure 2).

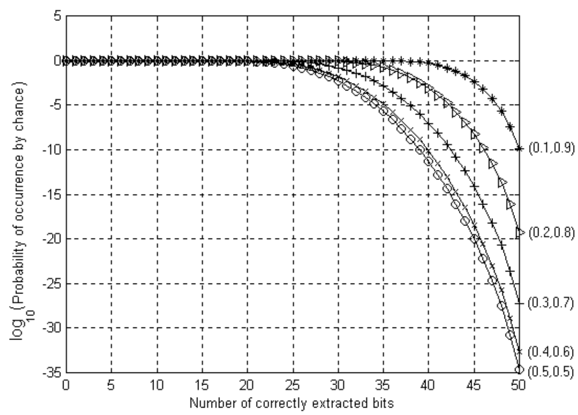

Figure 2 Probability of extracting certain number of correct bits due to pure chance for various distributions of 1's and 0's in the extracted 50-bit tuple. The probabilities of occurrence of 0's and 1's are indicated in the round brackets at right.

In the next section, we propose two general strategies how to obtain visual hash functions that are robust to a wide range of grayscale operations and some simple geometric transformations including rotation and scaling. The first method is based on the Fourier-Mellin transformation, and the second method utilizes a combination of a one-dimensional Fourier transform with a special scale independent functional of the image.

3. ROBUSTNESS TO GEOMETRIC TRANSFORMATIONS

The visual hash function described in the previous section is very robust with respect to grayscale modifications in which the geometrical location of pixels is not changed. However, a small amount of rotation or a slight change of scale can cause a rapid loss of correctly recovered bits. Also, operations, such as embossing (Laplacian filtering) prevent a successful extraction of bits. In this section, we present some new ideas how to construct robust hash functions that will return the same bit-strings from images that have undergone a combination of rotation, scale, and a wider range of grayscale operations including a Laplacian edge extractor. Two general methodologies are proposed.

While typical image processing operations can modify gray scales, colors, or the image histogram by a large margin, a human subject can easily relate the processed image to the original one. This is because the most important image characteristics are usually well captured with edges. Thus, we propose to use the image edges as a start for extracting bits in a robust manner. Since large differences in image brightness can undesirably influence the edge detector, we propose to preprocess the image before extracting the edges. We do so by normalizing the image contrast using histogram equalization. Thus, the first step of our new robust hash function for images is to normalize the histogram of the image I to obtain the contrast-normalized image $Norm\_Hist(I)$. As the next step, the Laplacian edge detector $L(\,)$ is applied to obtain the preprocessed image I''

$$I'' = L(Norm\_Hist(I)). \tag{2}$$

The normalized image I'' is less sensitive to grayscale operations than the original image. Another option is to use edge detectors based on identifying wavelet maxima. This algorithm gives consistent results even after operations of low-pass character[7]. Our task now is to make the hash independent of the orientation and scale of the image I''. Two different approaches are proposed in this section.

One possibility would be to use the Fourier-Mellin transformation [7] $\mathscr{FM}$ of I''. The transformation begins with a substitution of variables $(x, y) \rightarrow (\mu, \theta)$

$$x = e^{\mu}\cos\theta,$$
$$y = e^{\mu}\sin\theta.$$

In the new coordinate system (μ, θ), rotation and scale are transformed into a circular translation. We can further utilize the fact that the modulus of the Fourier transform is invariant with respect to circular spatial shifts (only the phase changes), which is a well-known property of the Fourier transform. Thus, taking the modulus of the Fourier transform in the new coordinate system will give us a function that is independent of rotation and scale. Therefore, it can be readil used for extraction of bits. The combined transformation is called Fourier-Mellin transformation[8] and it has been introduced in the field of digital watermarking for the first time by Ó Ruanaidh[9]. The complete robust hash function can therefore be written as

$$F_K(|\mathscr{F}\mathscr{M}(I'')|), \tag{3}$$

where F_K denotes a function with a parameter (key K. We require that f is a continuous function of its argument and a sensitive function of the paramete K. The latter could be arranged by inserting coefficients into the analytical expression of F and generate those coefficients from a pseudo-random number generator seeded with K.

The second approach to achieve robustness to rotation and scale is based on rotationally symmetrical patterns. We start by calculating the center of gravity of the edge map. Again, the wavelet maxima edge detecto [7] seems to work well for a large class of possible image distortions and the center of gravity seems to be reasonably invariant. We then rescale the image to a predetermined size (e.g., 512×512 pixels). To calculate the visual hash, we project the image onto a set of pseudo-random smooth patterns with circular symmetry centered around the center of gravity (see Figure 3). The projections are thresholded with an adaptive threshold as before.

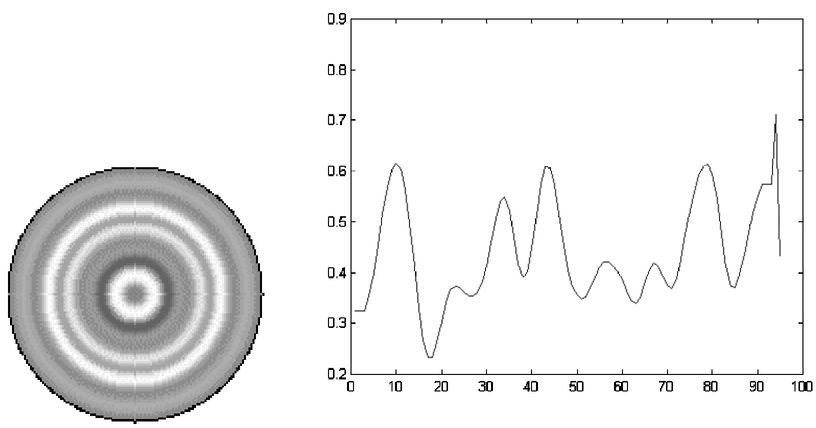

Figure 3 Example of a random circularly symmetrical pattern with its profile.

The disadvantage of this approach is that the circular patterns will in general be more correlated than two-dimensional random patterns, and as a consequence, the hash bits may exhibit strong correlations. This could be avoided b orthogonalizing the patterns prior to calculating the visual hash. Since it is necessary to do the orthogonalization in a one-dimensional space only, using the simple Gram-Schmidt procedure should produce a reasonably fast code. Another disadvantage of this approach is that the number of hash bits will most likely have to be lower than in the fully two-dimensional case.

The center of gravity of the edge map seems to be very robust to common image processing operations as well as small cropping as long as the cropped out part of the image does not contain edges. Rescaling the image to a predetermined size may seem dangerous if cropping should occur. Indeed, cropping an image will cause the visual hash malfunction. It appears that this problem has no easy solution because a cropped image is virtually a different image, and, therefore, it should produce a different hash. One possibility to overcome this problem would be to calculate the hash bits locally and produce a random spatial pattern as the result, rather than a set of hash bits. This spatial pattern could be the initial input for synthesizing a watermark for a specific watermarking scheme. This approach is attractive because it has the potential to provide a solution to robust oblivious watermarking that could survive non-linear random warping (StirMark).

4. GENERATING A WATERMARK USING THE HASH

Vast majority of watermarking schemes generate the watermark from a pseudo-random sequence. In this section, we explain how to synthesize a Gaussian sequence from N hash bits so that the pseudo-random sequence gradually changes with increased number of errors in the hash, yet sensitively depends on the secret key. In addition to that, we require that when approximately half of the hash is incorrect, the generated Gaussian sequence should not be correlated with the sequence produced from all 50 correct bits. To achieve this goal, we synthesize the pseudo-random Gaussian sequence b summing up uniformly distributed pseudo-random sequences obtained from a pseudo-random number generator (PRNG) seeded with a concatenation of the secret key, the block number (if the watermarking is done by blocks), and randomly chosen q-tuples of the extracted bits ($q \approx 5$). We start by generating q random permutations $\pi_1, \pi_2, \ldots, \pi_q$ of integers between 1 and N. The permutations could be fixed for all images and blocks or change with the block. Then for each i, $1 \le i \le N$, we seed a PRNG (with uniform probability distribution on $[-1,1]$) with a seed consisting of a concatenation of the secret ke K, the block number B, the number i, and q bits $\pi_1(i), \pi_2(i), \ldots, \pi_q(i)$. The PRNG then generates a pseudo-random sequence $\xi^{(i)}$ of a desired length (determined by the particular watermarking technique)

$$\xi^{(i)} = PRNG(K \oplus B \oplus i \oplus b_{\pi_1(i)} \oplus b_{\pi_2(i)} \oplus \ldots \oplus b_{\pi_q(i)}).$$

In the expression above the symbol $\oplus$ denotes concatenation. The final Gaussian sequence $\eta \in N(0,1)$ is obtained b summing up $\xi^{(i)}$ for all i and normalizing:

$$\eta = \sqrt{\frac{3}{N}} \sum_{i=1}^{N} \xi^{(i)}.$$

The process of generating the pseudo-random sequences $\xi^{(i)}$ is schematically depicted in Figure 4. If the probability of extracting 1 is the same as probability of extracting 0, we can easily estimate how many seeds will be recovered correctl for the correct secret key and similar blocks. If k bits out of N bits are recovered correctly, then approximately $(k/N)^q$ seeds (and consequently the sequences $\xi^{(i)}$) will be correct. If we use the wrong key or a dissimilar block, the number of correctly recovered seeds will be roughly $1/2^q$ which could be made much smaller than $(k/N)^q$ by choosing q appropriately (see Figure 5).

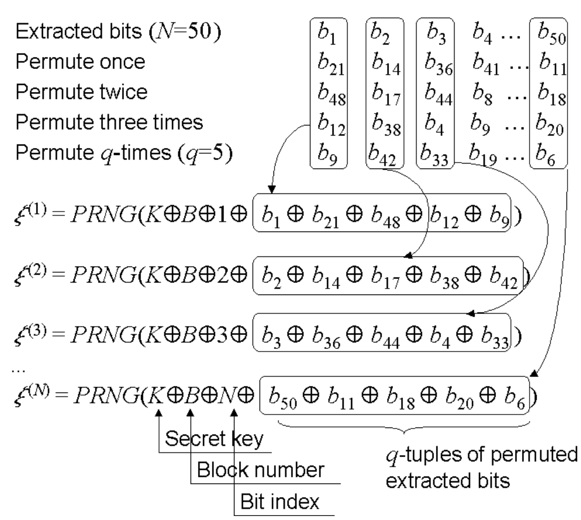

Figure 4 Synthesizing the Gaussian pseudo-random sequence from the extracted bits

In our simulations, we used q=5 as a compromise between the loss of correlation due to image degradation and creating a small correlation among dissimilar blocks for the same secret key and the same fixed block. The watermarking scheme that utilizes our robust bit extraction technique is schematically depicted in Figure 6.

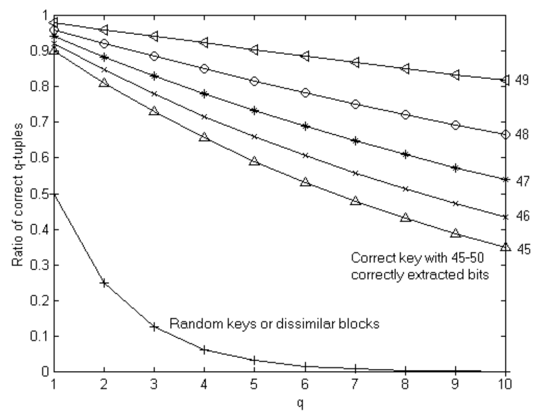

Figure 5 Ratio of correct seeds (q-tuples) as a function of k (the number of correctly recovered bits) and the factor q

5. CONCLUSION AND FUTURE DIRECTIONS

In this paper, we describe an algorithm for robust visual hash functions with applications to digital image watermarking for authentication and integrity verification of video data and still images. The robust image digest can also be used as a search index for efficient database searches. The hash function depends on a parameter K (a secret key) in a sensitive manner and on the image in a robust, continuous manner. The hash function always returns the same N bits from an image of arbitrary size. The bits obtained from two different images or for two different keys K will generally be different (uncorrelated). However, for the same key K, two images that can be matched after applying gray scale operations, such as lossy compression, recoloring, filtering, noise adding, gamma correction, and simple geometrical operations including rotation and scaling, the extracted N-tuple will be almost the same. We also explain how the extracted N-tuple can be further utilized for synthesizing a Gaussian sequence that gradually changes with increasing number of errors in the extracted bits. Thus the robust hash function can be used for generating pseudo-random watermark sequences that depend sensitively on a secret key yet continuously on the image. This robustness enables us to construct watermarks that depend on the original unwatermarked image in a non-trivial manner while making it possible to recover the watermark without having to access any information about the original image (oblivious watermarking). Such watermarks play an important role for authenticating videos or still images taken with a digital camera [3].

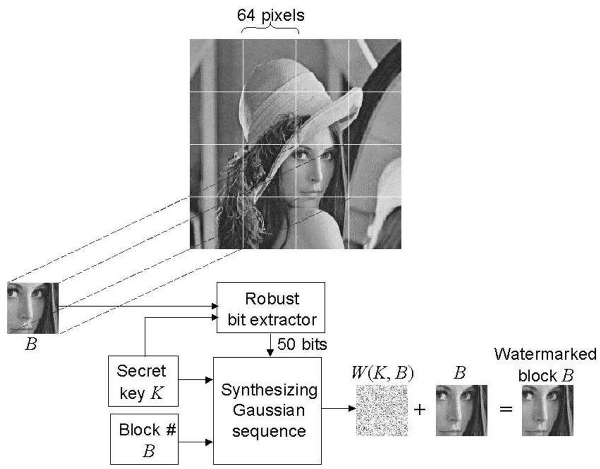

Figure 6 General scheme for watermarking using the proposed bit extraction procedure

In those watermarking schemes that are robust with respect to rotation and scale, the watermark recovery is typicall obtained using special synchronization patterns used to detect the geometric operations applied to the image [9]. The watermark is read after those geometric operations have been undone. Since the accuracy with which the geometrical operations are detected is usually limited, it is necessary to have a robust technique for generating the watermark at the receiving end. For current robust hash functions[6], even small rotations and change of scale can produce large differences in the extracted bits. The methods proposed in this paper constitute a step towards robust message digest functions that are insensitive to rotation and scale. The first method is based on Fourier-Mellin transformation combined with a robust edge detector. The second technique works with DC-free circularly symmetrical random smooth patterns centered on the center of gravity calculated from the edge map.

The visual hash produces different hash bits for cropped images after a substantial amount of edges have been cropped out. It appears that this problem has no easy solution because a cropped image is virtually a different image, and, therefore, it should produce a different hash. One possibility to overcome this problem would be to calculate the hash bits locally and produce a random spatial pattern as the result, rather than a set of hash bits. This spatial pattern could be the initial input for synthesizing a watermark for a specific watermarking scheme. This approach is attractive because it has the potential to provide a solution to robust oblivious watermarking that could survive non-linear random warping (StirMark). This possibility will be investigated as part of our future research.

The future effort will concentrate on design specifics of the proposed hash functions and testing the performance on real imagery. The performance of the proposed robust hash functions is directly influenced by the function f_K and the circula patterns. The sensitivity with respect to small modifications of their arguments will ultimately determine the overall properties of the hash function. In the future, we intend to investigate the best choice of those functions to maximize the robustness.

As another application of robust hash functions, we mention indices for efficient image database search. There are man quantities that could be derived from images using which one can search a database in an efficient manner. Man indices are based on color information that can be extracted from a histogram. However, such indices are not useful if the image has been processed using histogram equalization, or recolored. The essence of an image can be well captured using its edges. Our method captures the mutual spatial relationship among edges rather than color information. This relationship is independent of the image orientation and size and on typical non-destructive image processing operations, such as recoloring, brightness adjustment, filtering, lossy compression, or small noise adding. Thus, it is computationall much more efficient to search an extensive image database by matching the extracted bit-string rather than whole images.

ACKNOWLEDGEMENTS

The work on this paper was supported by Air Force Research Laboratory, Air Force Material Command, USAF, under a Phase II SBIR grant number F30602-98-C-0049. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, eithe expressed or implied, of Air Force Research Laboratory, or the U. S. Government.

REFERENCES

1. M. D. Swanson, B. Zhu, and A. H. Tewfik, "Data Hiding for Video in Video", *Proc. ICIP '97*, vol. II, pp. 676–679, 1997.
2. M. Holliman, N. Memon, and M. M. Yeung, "On the Need for Image Dependent Keys for Watermarking", *Proc. Content Security and Data Hiding in Digital Media* , Newark, NJ, May 14, 1999.
3. J. Fridrich, "Methods for Tamper Detection in Digital Images", *Proc. ACM Multimedia 1999, Workshop on Multimedia and Security*, October 30 – November 5, 1999
4. I. J. Cox and J.-P. M. G. Linnartz, "Public watermarks and resistance to tampering", *ICIP'97*, Santa Barbara, California, October 1997. Paper appears only in CD version of proceedings.
5. I. J. Cox and J.-P. M. G. Linnartz, "Some general methods for tampering with watermarks", preprint, 1998.
6. J. Fridrich, "Robust Bit Extraction From Images", *Proc. ICMCS'99*, vol. 2, Florence, Italy, June 7–11, pp. 536–540, 1999.
7. S. Mallat, *A Wavelet Tour of Signal Processing*, Chestnut Hill, MA, Academic Press, 1998.
8. R. D. Brandt and F. Lin, "Representations that uniquely characterize images modulo translation, rotation and scaling", *Pattern Recognition Letters* **17**, pp. 1001–1015, 1996.
9. J. J. K. Ó Ruanaidh and T. Pun, "Rotation, scale and translation invariant digital image watermarking", *Proc. of the ICIP'97*, vol. 1, pp. 536–539, Santa Barbara, California, 1997.