# ROBUST HASH FOR DETECTING AND LOCALIZING IMAGE TAMPERING

*Sujoy Roy* *Qibin Sun*

Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore

## ABSTRACT

An image hash should be (1) robust to allowable operations and (2) sensitive to illegal manipulations (like image tampering) and distinct queries. Although existing methods try to address the first issue, the second issue has not been adequately looked into, primarily the issue of localizing tampering. This is primarily because of the difficulty in meeting two contradictory requirements. First, the hash should be small and second, to detect localized tampering, the amount of information in the hash about the original should be as large as possible. The desynchronization of the query with the original further aggravates the problem. Hence a tradeoff between these factors needs to be found. This paper presents an image hashing approach that is both robust and sensitive to not only detect but also localize tampering using a small signature ($< 1kB$). To our knowledge this is the first hashing method that can localize image tampering using a small signature that is not embedded into the image, like in watermarking.

***Index Terms***— Locality preserving hashing, edge histogram, local region descriptors.

## 1. INTRODUCTION

An image hash is a short signature of the image that preserves its semantic information under allowable changes made to it while at the same time differentiates it from a different image (either distinct or tampered). That is, it should be robust to allowable modifications (like small rotations, compression, scaling, addition of noise etc) and sensitive to distinct images or illegal manipulations to the original like tampering. Hashes find application in verifying the authenticity of protected content. Figure 1 depicts an original image and its manipulated copy (slightly rotated, cropped, JPEG compressed, stretched and locally tampered). Given the hash of the original and the tampered copy, the goal of the hashing method is to verify the authenticity of the query. Only allowably modified images are declared authentic. Tampered or distinct images are declared non-authentic. Furthermore, the system may require the ability to localize the tampered region in tampered images.

A typical image hashing method consists of two steps: (1) hash generation and (2) verification. For hash generation, a set of features $I \in \{\mathbf{R}^n\}$ is extracted from the image and a function $f : I \mapsto h$, maps (also called bit extraction process) them to a bit sequence $h \in \{0,1\}^L$, where $\{\mathbf{R}^n\}$ denotes a set of vectors in $n$ dimensional real space, $\{0,1\}^L$ denotes a bit sequence of size $L$. If $|I|$ denotes the size of $I$ and $bit(x)$ the bit representation of any real number $x$, $|I| \times n \times bit(x) \gg L$, i.e., $h$ is a short hash of $I$. Under any noise $N$, $I$ gets changed to $\tilde{I}$. During verification, given $h$ and $\tilde{I}$, the detector decides whether $\tilde{I}$ is authentic or not. Verification is done by computing the hash $\tilde{h}$ of $\tilde{I}$ and comparing it with $h$ based on a dissimilarity/similarity measure. The noise $N$ can be allowable modifications (affects the robustness) or illegal manipulations (that can affect both robustness and sensitivity).

(a) (b)

**Fig. 1**. (a) Original Image (b) Illegally tampered image (also rotated by $2^o$, cropped, stretched, JPEG compressed (Q=20)).

Existing image hashing methods (that primarily address the issue of robustness) can be categorized as belonging to (1) exhaustive search based[1] and (2) robust representation based approach[2, 3, 4, 5, 7]. In an exhaustive search based approach, the noise $N$ is modeled by some fixed distortion model (e.g., affine transform) and the hash $h$ carries some alignment information about the original. For verification, the right alignment between $I$ and $\tilde{I}$ is searched for by trying all possible values for the parameters of the noise model $N$, reverse applying on $\tilde{I}$ and comparing the hash $\tilde{h}$ of $\tilde{I}$ with $h$ using a similarity measure. If a very close alignment is indeed found, the query is declared authentic. On the other hand, in a robust representation based approach, a hash $h$, robust to noise $N$ (say RST, compression, additive noise etc), is generated from the original. During verification, the hash $\tilde{h}$ of the query $\tilde{I}$, is generated and compared with $h$ using a similarity measure. If $h$ is similar to $\tilde{h}$, the query is declared authentic.

The above approaches have their advantages and disadvantages. Exhaustive search based methods clearly suffer from impractical levels of search complexity, although in theory

they can synchronize the query with the original for effective verification performance. Lack of content information as part of the hash also leads to high false positive detection error[6]. On the other hand, in a robust representation based method although the hash $h$ carries robust content information, desynchronization of the query with respect to the original and lack of alignment information as part of the hash significantly limits the verification performance. This makes it clear that both alignment and robust content information should be part of an effective signature based method although this can significant increase the signature size. This is particularly essential for localizing tampering in images using a signature based approach.

The problem of localizing tampering in images can also be solved using a watermarking based approach, wherein, a watermark is inserted into the image at the point of creation, and during verification, the watermark is extracted to verify if there was any allowable modification or illegal manipulation performed on the image. Any tampering can be localized from the damage to the watermark. A clear disadvantage in using watermarking is the need for distorting the content. In the case of image hashing for authentication (or a signature based approach), the signature is associated with the image as header information and must be small. It is particularly difficult to localize tampering using a signature based approach because of having to meet two contradictory requirements. First, the signature (hash) should be small and second, to detect localized tampering, the amount of information in the hash about the original, both content and alignment, (as realized from the analysis of existing approaches above) should be as large as possible. Therefore, a tradeoff between these two contradictory requirements needs to be found. To resolve this forms the motivation for this work.

This paper proposes a novel signature based approach for localizing tampering in images, wherein the signature carries both content and alignment information and at the same time is short in size ($< 1kB$). The proposed method builds on and incorporates some of the significant advantages of the hashing method proposed in [7]. Some of the disadvantages of the method [7], namely detecting local tampering has been alleviated in the proposed method. In fact, the proposed method goes further in not only detecting but actually localizing image tampering with a small sized hash. The next section gives the formulation of the proposed approach.

## 2. FORMULATION

**Hash Generation** Given a set of features $I = \{F_1, \ldots, F_m\}$, $F_1 \in \mathbf{R}^n$, design a function $f$ to generate a bit sequence $h$, where $h \in \{0,1\}^L$, such that $\{0,1\}^L \ll \mathbf{R}^{m \times n}$ in terms of bits.

**Verification** During query verification, given a query, as a set of features $\tilde{I} = \{\tilde{F}_1, \tilde{F}_2, \ldots, \tilde{F}_k\}$ and the hash $h$, verify whether $\tilde{I}$ is an authentic version of the original set $I$. Note that $I$ and $\tilde{I}$ are not synchronized and their sizes need not be the same. The verification routine ascertains the authenticity of the query. A query is declared authentic only if it is an allowably modified version of the original. Under illegal manipulations like localized tampering, the verification routine localizes the tampered region in $\tilde{I}$. The next section describes the proposed hashing method.

## 3. PROPOSED METHOD

**Hash Generation** The hash generation process is a simple extension of the hashing method proposed in [7]. It consists of a feature extraction step, followed by a bit extraction process that generates a bit sequence $h$ of fixed size. The final bit sequence is actually a combination of two bit sequences $h_1$ and $h_2$, which are generated independently. The process of generation of $h_1$ and $h_2$ are described herein.

**Generation of $h_1$:** The image is first down-sampled and then a set of features $I = \{F_1, \ldots, F_m\}$ with $F_i \in \mathbf{R}^{128}$ are extracted from it. The feature $F_i$ is a local region descriptor[8], which has been shown [9] to be robust to several geometric transformations. Next, a binary representation of $F_i$ is computed in the bit extraction process, which entails the following steps: (1) take a random hyperplane $H \in \mathbf{R}^{128}$ (generated by a known secret key), that passes through the centroid of the feature distribution, (2) label the feature vectors on either side of the hyperplane as 0 or 1 depending on whether they lie on the left or right side of the hyperplane. Continue steps (1) and (2) for $d$ such random hyperplanes. Figure 2 depicts the labeling process for two hyperplanes. This process maps each $F_i$ to a sequence of bits $t_i \in \{0,1\}^d$. Location (2D) information for 3-5 feature points are also added to the hash information. Hence the size of the final bit sequence $h_1 = t_1 \oplus t_2 \oplus \ldots t_m + \ell$ is $md + \ell$, where $\oplus$ is the concatenation operator and $\ell$ is the 2D location information (in bits) for the most stable 3-5 feature points.

**Generation of $h_2$:** The image is first downsampled and filtered using an anisotropic diffusion filter and then edge detection is performed to generate its edge image. The orientation of the edges in the edge image are quantized to directions in $[0, 45, 90, 135, 180]$. Next the edge image is divided into non-overlapping blocks and the edge histogram for each block is computed. The edge histograms of each block are concatenated to generate $h_2$. Each edge histogram is represented by 15 bits For an image which is divided into 16 blocks, this would generate a hash $h_2$ of size 240 bits.

**Verification** The verification stage uses a localized threshold to do pairwise matching of the bit sequences which are part of the hash component $h_1$. This gives pairs of corresponding 2D points in the original and query. Now 2D point location information as included in $h_1$ of some of the points in the original can be used to find the mapping transformation.

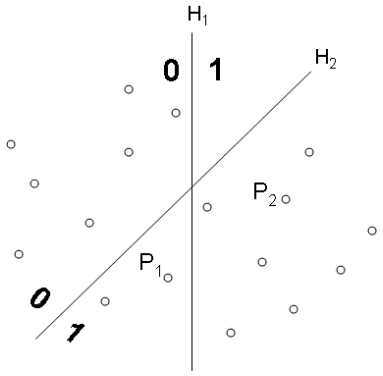


**Fig. 2**. Illustration of the intermediate bit sequence generation process. For two hyperplanes $H_1$ and $H_2$, the points $P_1$ and $P_2$ are binarized as $\{01\}$ and $\{11\}$ respectively.

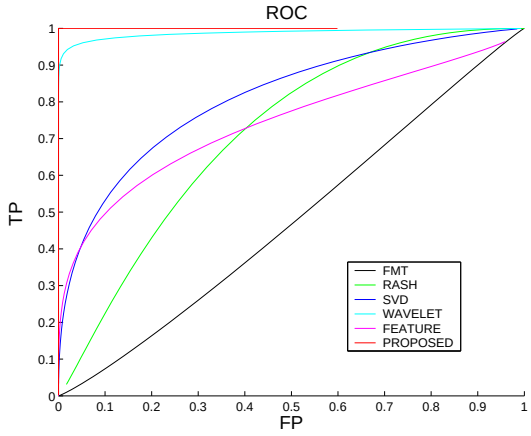


**Fig. 3**. ROC curve comparing the performance of the proposed method with existing methods for images rotated ($20^o$), cropped ($30\%$) and JPEG compressed ($QF = 10$) against 50 distinct images from the USC-SIPI database.

Note that for an affine model, only 3 correspondences are sufficient for computing the parameters. The query is then inverse transformed using the computed transformation parameters (based on the chosen model, affine, projective etc) and aligned with the original. Next, the aligned query is preprocessed as discussed before, divided into blocks and the edge histogram of each block is extracted and compared with the hash component $h_2$ (based on a threshold $T$) of the original image. Image blocks with dissimilarity value greater than $T$, indicate a probable tampered block. Note that the resolution of tamper localization depends on the block size chosen. A smaller block size will increase the hash size, while improving the tamper localization resolution. In our implementation, the image was first downsampled and then divided into 16 blocks. Next tampered blocks are localized.

## 4. EXPERIMENTS AND ANALYSIS

For our experiments a collection of 50 dissimilar images from the USC-SIPI database was used. Some of these images where tampered by performing splicing, content removal, and content rearranging, to generate perceptually indistinguishable image queries. Performance of the method just based on $h_1$ was ascertained based on comparison with three existing imaging hashing methods in terms of robustness-false alarm trade-off effectiveness, under allowable transformations.

The hash component $h_1$ helps to discriminate between allowably transformed and distinct queries, using a localized threshold[7] and also align the query. Values of $m = 50$ and $d = 30$ were used in [7] to generate a hash of size $h_1 = 1500$ bits. It is noted that for higher value of $d$ (say $d = 60$), we get higher discriminative ability. In that case, the number of feature points considered, $m$, can be reduced significantly. In our implementation, 10 stable region descriptors were chosen as features, along with the 2D coordinate information for 5 most stable amongst them, represented in bits. Thus, for $m = 10$ and $d = 60$, $h_1$ requires $md = 600 + 80 = 680$ bits. On the other hand $h_2$ requires 240 bits. Hence the total size of the hash $h$ is $L = 920$ bits ($< 1kB$).

Figure 3 depicts the ROC curves for 5 state-of-the art methods based on Fourier-Mellin invariants (FMI)[2], radial basis projections(RASH) [3], wavelets[5], SVD[4], structure matching (Feature) [1] compared to the proposed method. The hash component $h_1$ was used in this comparison. The two classes compared are similarity between the 50 distinct images and the similarity of the original images with its modified versions, rotated about a point $[100, 100]$ pixels away from the center by 20 degrees, cropped by $30\%$ and JPEG compressed to quality factor 10. Note that the proposed method clearly achieves very high discrimination ability compared to the other methods. This is due to the high discrimination capacity of SIFT features which is preserved by the locality preserving hash based bit extraction step. Furthermore, the hash $h_2$ complements $h_1$ by allowing for detection and localizing of tampering, which is not possible in the other methods.

Figure 4 depicts an example of the effectiveness of our proposed method in localizing image tampering. Note that specifically for this example there was no introduction of external content in the query. Therefore simple robust content representation based hashes without structure information would fail to even detect any change in the query. For localizing, first $h_1$ is used to register and align the query with the original. Next $h_2$ is used to detect and localize any local tampering. The proposed method can detect any form of tampering, namely, insertions, deletions, exchange of patches within the same image etc. The idea is that any intentional tampering leaves behind significant addition or deletion of


(a) (b) (c)

**Fig. 4**. (a) Original Image (b) Illegally tampered image (c) Tampering detected.

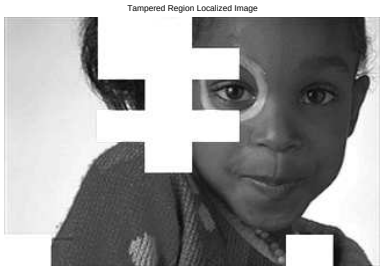


**Fig. 5**. Tamper localization of example in Figure 1.

content information, primarily edge boundary information. The resolution of the patch detection depends on the size of the image blocks considered and hence affects the hash size. Figure 5 depicts the localization of tampering for the image in Figure 1.

## 5. DISCUSSION

The proposed method can be seen as a unified method that combines the advantages of an exhaustive search based hashing and robust representation based hashing methods. The locality preserving projection of region descriptors can be seen as a short robust representation whereas the availability of 2D point location information useful for aligning the original with the query is a component of an exhaustive search based hashing method. The availability of content information as a robust bit representation helps in reducing the search complexity and decreasing false positive error, both of which are drawbacks of an exhaustive search based method. On the other hand availability of point location information as part of the hash helps in registering the query with the original which in turn addresses the synchronization problem. Once aligned, the use of edge histogram information after some preprocessing of the query allows localizing any tampering. As per our knowledge none of the existing hashing methods solve the problem of detecting and localizing image tampering.

## 6. REFERENCES

[1] V. Monga, D. Vats, and B. L. Evans, "Image authentication under geometric attacks via structure matching," in *ICME*, July 2005.

[2] Ashwin Swaminathan, Yinian Mao, and Min Wu, "Robust and secure image hashing," *accepted by IEEE Transactions on Information Forensics and Security*, to appear June 2006.

[3] C. De Roover, C. De Vleeschouwer, F. Lefebvre, and B. Macq, "Robust video hashing based on radial variance projections of key-frames," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 4020–4037, October 2005.

[4] S. S. Kozat, K. Mihcak, and R. Venkatesan, "Robust perceptual image hashing via matrix invariances," in *Proc. IEEE Conf. on Image Processing*, Oct, 2004.

[5] M. H. Jakubowski R. Venkatesan, S.-M. Koon and P. Moulin, "Robust image hashing," in *Int. Conf. Image Processing, Vancouver, Canada.*, September, 2000.

[6] J. Lichtenauer, I. Setyawan, T. Kalker, and R. Lagendijk, "Exhaustive geometric search and false positive watermark detection probability," in *Proc. SPIE Security and Watermarking Multimedia Contents V*, Jan, 2003, pp. 203–214.

[7] S. Roy, X. Zhu, J. Yuan, and E-C. Chang, "On preserving robustness false alarm tradeoff in media hashing," in *Proc. SPIE Visual Communication and Image Processing (to appear)*, Jan, 2007.

[8] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[9] Krystian Mikolajczyk and Cordelia Schmid, "A performance evaluation of local descriptors," in *International Conference on Computer Vision & Pattern Recognition*, June 2003, vol. 2, pp. 257–263.