Digital watermarking techniques: A case study in fingerprints and faces

Sonia Jain
Department of Electrical Engineering
Princeton University
jain@princeton.edu

Abstract

Watermarks have been used for centuries to authenticate images and prevent forgery. Today watermarks are embedded into digital images so that legitimate owners can assert ownership and ensure the validity of their data. However, these images (like fingerprints) are being transmitted over channels with increased frequency, thus the potential for attacks during transmission is a major concern. Recipients need a mechanism which can verify the integrity of their image. If the images are watermarked, it is straightforward for the owner to authenticate the images. But, determining whether the correct image has in fact been sent and that it has been transmitted without alteration requires use of data the owner would prefer not to transmit (namely the watermark itself or the original image). Therefore, a different key needs to be used by the receiver for authentication. We propose a local average scheme where an executable compares the block by block local average of the transmitted image and the received image. This is robust enough to detect even the most minor changes and determine where such changes took place in the image.

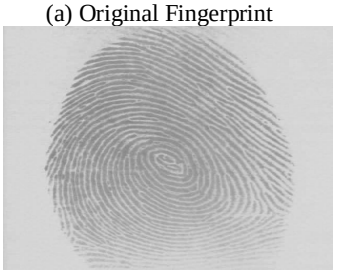
1 Introduction

A method for establishing the identity of an individual is essential in all transactions whether they are commercial or personal. The ability to establish identity with certainty can prevent fraud or forgery. In the midst of an electronic revolution, this remains a major concern in e-commerce, telecommunications, healthcare, and security. While verifying the identity of individuals has always been a concern, in the past, it was primarily handled with information such as a social security number or a password, or some sort of physical key like an ID card. However, these methods are gradually losing favor and are being replaced by biometrics parameters, such as fingerprint, speech and iris, which are "unique" to an individual and so they can not be easily altered [1].

Watermarks have long been used for authentication and to prevent fraud and forgery. Most corporate stationary bears an embedded logo, as does much of the world's paper currency. These watermarking schemes are fairly simple and serve their purpose efficiently. Reproducing watermarked bills is a challenge for the average criminal. Considering their success to date, applying (digital) watermarks to biometric data and identifiers is a plausible way to determine their ownership and deter their tampering [2]. However, unlike the previous use of watermarks, for biometric applications, we are not as interested in visible watermarks as invisible watermarks. Invisible watermarks, as their name indicates, do not appear to visually affect the data that they are embedded in. This method is desired if one does not want to perceptually alter the image. While the images in Figures 1 (a) and (c) and Figures 1 (b) and (d) look identical, the images in Figures 1(c) and (d) have been watermarked using the digital watermarks shown in figures 1 (e) and (f), respectively.

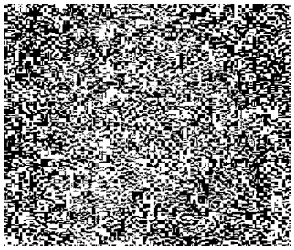
(a) Original Fingerprint

(b) Original Face



(b) Watermarked Fingerprint

(d) Watermarked Face



(e) Watermark

(f) Watermark

Figure 1: Examples of Invisible Watermarks

A number of biometrics are in use today. Signatures are the most commonly used biometric identifiers. The growing use of pressure sensitive devices to capture signatures on credit cards receipts in stores implies the creation of large databases containing digital versions of customer's signatures [3]. Work has been done to successfully watermark these identifiers [4]. Unfortunately, signatures of the same individual are highly variable and easily forged. Therefore, signature-based identification cannot assure high accuracy. Fingerprints present themselves as a unique biometric that can be used to immediately establish personal identity [1]. However once the fingerprints are digitized, stored, and transmitted over a network, they become susceptible to malicious as well as accidental attacks. In order to preserve the fidelity of this information and prevent alterations from being made at will, a protective scheme must be used. Protection of biometric identifiers is even more important due to their uniqueness. Unlike passwords or a PIN, which can be changed if compromised, no such re-issuing policy is available for biometric identifiers. Thus, the need to protect them is even greater.

One potential danger with sensitive databases containing biometric identifiers is that they are likely to be attacked by hackers or criminals. Watermarking the information in these databases can allow the integrity of the contents to be verified. But another danger is that this critical data can be attacked while it is being transmitted. For example, a third party could intercept this data and maliciously alter the data before re-transmitting it to its final destination. The transmission problem is even more critical in cellular and wireless channels. The channels themselves are quite noisy and can degrade the signal quality. Additionally, data transmitted through wireless channels is far from secure as they are omni-directional, and as such can be eavesdropped with relative ease. The growth of the wireless market and e-commerce applications for PDAs requires a robust method for data security. There are compact solid state sensors already available in the market, which can capture fingerprints or faces for the purpose of identity verification. These devices can also be easily attached to PDAs and other hand-held cellular devices for use in identification and verification. Considering all the noise and distortion in cellular channels, a method for introducing a watermark would be useful, as it would help us determine whether the image had been substantially altered or tampered with [5].

2 Watermarking Techniques

The purpose of watermarks is two-fold: (i) they can be used to determine ownership, and (ii) they can be used to detect tampering. There are two necessary features that all watermarks must possess. First, all watermarks should be detectable. In order to determine ownership, it is imperative that one be able to recover the watermark. There are essentially two mechanisms by which a watermark can be recovered. Incomplete watermarks can only be recovered provided the original image is available. Complete watermarks can be recovered regardless. Complete watermarks are more desirable as they apply to a broader spectrum of applications. When watermarking large files or a large number of files in a database, complete watermarks are preferable as they make it unnecessary to store multiple copies of the original (unwatermarked) file. Second, watermarks must be robust to various types of processing of the signal (i.e. cropping, filtering, translation, compression, etc.). If the watermark is not robust, it serves little purpose, as ownership will be lost upon processing. However, having some built-in fragileness can be useful at times. If fragile watermarks are used and the data is altered, the watermark can pinpoint the areas that were changed. Fragile watermarks can detect minor changes or tampering of data. Robust watermarks on the other hand, are useful for detecting large-scale attacks on data.

Many different watermarking schemes have been developed. One of the first watermarking algorithms involved manipulating the least significant bit (LSB) of pixels in the spatial domain [6]. There are several ways in which these LSB schemes can be applied. For example, either all the LSBs could be altered, or a randomly chosen set of LSBs could be altered. These schemes are particularly useful because of their fragileness. If one alters an image, it is more than likely that the LSB will be altered as well. Unfortunately, this same fragileness can cause a host of other problems. It makes it possible to obliterate the whole watermark. If a sufficient number of LSBs are altered, the watermark becomes unrecoverable. Also, it is possible to alter an image without changing the LSBs. If this is done, the watermark essentially serves no purpose, as it cannot be used to detect tampering.

In general, spatial (pixel) domain schemes are too fragile to withstand an attack. As a result, frequency domain solutions have been developed. There are two general frequency domain algorithms, a spread spectrum method and a block method. The spread spectrum method has been discussed at length by *Cox et al.* [7]. Essentially, the DCT of the entire image is taken, and the watermark is applied to pre-selected frequencies. If the image DCT is represented by $V(j, k)$, and the watermark is

W(j, k), then the watermarked image is $V^*(j,k) = V(j,k) + \alpha W(j,k)$, where *W(j, k)* has the normal distribution and α is a scaling parameter. In the simple version of the method, the value of alpha is set to 0.1. For better results, α can be derived from the JND (just noticeable difference) matrix. The JND matrix contains the average value that can be added to each pixel without causing a noticeable perceptual change in the image.

Another method that is often used is the block DCT method. It is similar to the spread spectrum method, however instead of taking the DCT of the entire image, the DCT is taken for 8x8 (or 16x16, etc.) blocks. This method allows the localization of watermarks. It also has the advantage since it is compatible with lossy compression techniques like MPEG. It could be built straight into an MPEG processor. However, it has its disadvantages. Since it is applied to each image segment separately, it is easier to remove this type of a watermark [8].

Given the popularity of the JPEG format, developing watermarks that are robust to compression is a major concern. There are several watermarking algorithms that are robust to compression. In these schemes the watermark is often inserted into the frequency domain of the compressed image. When watermarks are placed into uncompressed images, the watermark may be damaged upon compression. In fact, it can be damaged to a point where it is no longer recognizable. But by inserting the watermark in the compressed frequency domain, compression should have little effect on the watermark (provided that the attacker does no compress the image to a level much below where the watermark was placed) [9].

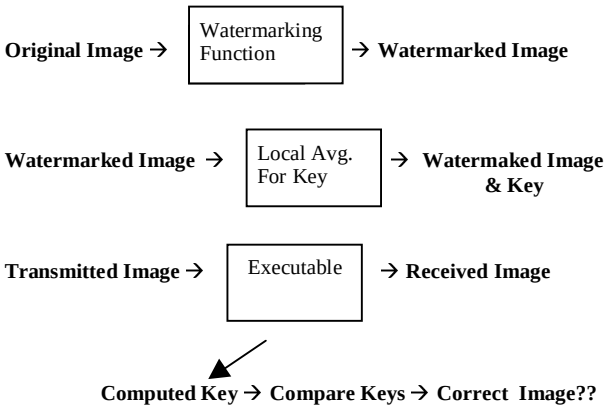
Another embedding scheme that is often discussed involves placing multiple watermarks into a single image. By placing multiple watermarks into the same image, the ability to determine *whether* an image has been tampered with and *where* it has been tampered with increases. In general, two watermarks are placed into an image. One of the watermarks is robust to image processing and the other accurately detects minor changes in the images (i.e., it is fragile). In previous studies, a watermark was inserted into the frequency domain (robust watermark) as well as into the pixel domain (fragile watermark) [10]. There are a couple of drawbacks to this sort of scheme. Since two watermarks are being inserted, neither can be at its maximum value. It is imperative that their magnitudes be scaled down. Also when inserting the watermarks it is essential to insert the robust watermark first [11]. If the fragile watermark is inserted first, then as soon as the robust watermark is inserted, the fragile watermark will detect non-existent tampering!

3 Proposed Technique

There are various watermarking schemes in use today, and for each scheme there are straightforward mechanisms by which tampering can be detected and ownership determined. However, often times in order to do so, some knowledge exclusive to the owner must be used. There are cases where the owner of an image may wish to transmit it to others. In such cases, how is the receiver to determine whether the image received has not been damaged? Clearly, some sort of *key* must be used.

On a very elementary level, the watermark or the original image itself could be used as a key. However, either of these would be a poor choice. The watermark is akin to a PIN or password. Allowing others access to it could be detrimental as they could ascribe and remove your ownership from images. Transmitting the original image itself, defeats the purpose of a watermark as now the receiver could choose to watermark the original image with a different watermark, transferring ownership to them.

Ideally, the key being used should be unique for each image, this can prevent a clever hacker from making attacks. An idea that comes to mind is the block by block characterization of images. One potential key that could be used is a matrix of local averages. The local average of the image to be transmitted could be computed in (say) 5x5 blocks and used as a key. The receiver could compare the key to a set of local averages computed on the received image. Thus, the authenticity of the image can be easily established. This key is attractive since it would be a fraction of the size of the original image, and in addition it could pinpoint to within a pixel's neighborhood where the changes occurred. It does have a disadvantage in that an expert hacker could alter an image without changing the local averages.

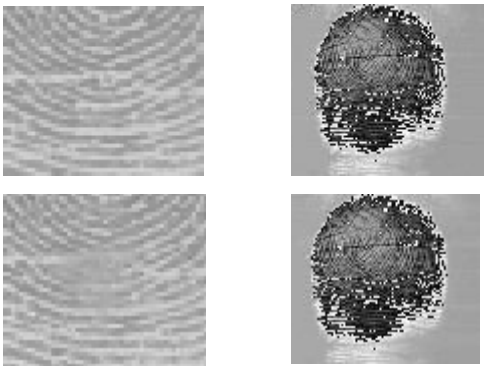


4 Experimental Results

The local average technique was used to detect image tampering in four specific scenarios: (i) smudging, (ii) compression, and (iii) Wiener filtering for a number of images. An executable was transmitted to enable the receiver to produce a key for the image, and compare it to the actual key. The keys are essentially amplified versions of 5x5 block local averages. The magnification makes small differences easier to detect, which enables in the detection of operations such as compression.

However, it also tends to increase the weight of noise effects, like transmission noise. For that reason, a threshold needs to be established which should be a function of the key magnification.

Smudging is one type of image alteration that was examined. It was easily detected in both fingerprints and faces using the local average based key. In fact, looking at the actual key and the generated key, it was almost obvious where the alteration had occurred. But if unconvinced, the user could compare the keys numerically, and determine where the image had been altered and by how much.



(c) above, (a) at top (d) above, (b) at top

Figure 2. Detecting smudging in a fingerprint image. Figure (c) shows a magnified version of the area that was smudged. The original can be seen in figure (a). Figures (b) and (d) show the keys for figures (a) and (b), respectively.

Image compression is more difficult to detect using this technique. In fact, the human eye can barely differentiate between an uncompressed image and an image that has been compressed up to a 75% quality level. The local average scheme is capable of detecting compression if it is at a level of 70% or below. At the 90% compression level, the plot of the difference in the keys indicates that there have been some changes made in the original image. However, it is unclear as to whether the change is simply due to transmission noise or malicious alteration. Detecting compression at high quality levels is not as crucial since it rarely causes major changes in the image. Even at the 70% level, it is difficult to assess which is the compressed image and which is the uncompressed image.

Another fairly common form of image processing is filtering. Wiener filters and other related filtering schemes have the ability to smooth images. The resulting smoothing alters almost every pixel to some degree. In our experiments, the entire image was Wiener filtered, though it is possible to simply filter sub-sections of the image. The smoothing was detectable perceptually, even though it was performed in small steps (10x10 pixels). Numerically, it was even more apparent, as visualized in the graph below.

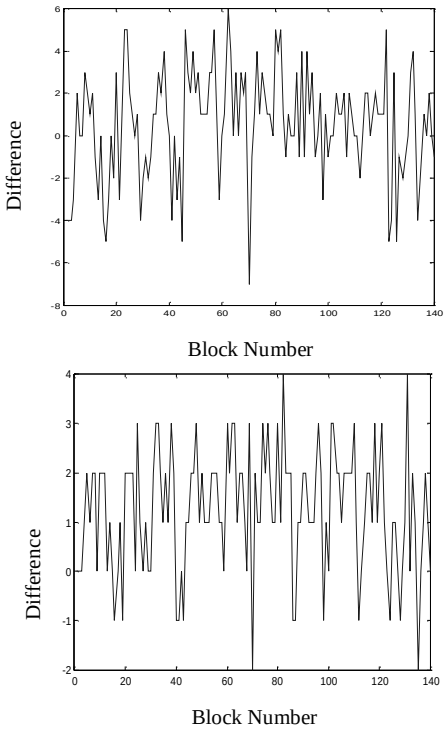


Figure 3: Detecting image compression in fingerprint images. The graphs show the difference between the keys, block by block. The top most graph is a plot for the 70% compression level, and above is a plot of the 90% compression level.

Another fairly common form of image processing is filtering. Wiener filters and other related filtering schemes have the ability to smooth images. The resulting smoothing alters almost every pixel to some degree. In our experiments, the entire image was Wiener filtered, though it is possible to simply filter sub-sections of the image. The smoothing was detectable perceptually, even though it was performed in small steps (10x10 pixels). Numerically, it was even more apparent, as visualized in the graph below.

(c)above, (a) at top (d) above, (b) at top

Figure 4: Detecting image smoothing. Figure (a) is the original watermarked image and figure (b) is the same image except that it has been filtered (smoothed). Figures (c) and (d) are the keys produced for the unaltered and the altered image, respectively.

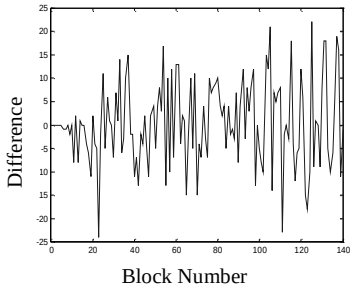


Figure 5: Difference between the two keys of figures 4(b) and (d) block by block. There are some huge differences!

5 Conclusion

Watermarking biometric data is a still a relatively new issue, but it is of growing importance as more robust methods of verification and authentication are being used. Biometrics provide the necessary unique characteristics but their validity must be ensured. This can be guaranteed to an extent by watermarks. Unfortunately, they can not provide a foolproof solution especially when the transmission of data is involved. A receiver can not always determine whether or not she has received the correct data without the sender giving her access to critical information (i.e., the watermark). Therefore, a key needs to be used. The key proposed here is one of many potential methods. The local average scheme creates a semi-unique key for each data set transmitted and thus is harder to tamper with. It also has the ability to pin-point where tampering has occurred up to a small pixel window. Thus data security can be assured in databases as well as in transmission.

However, it is only a semi-unique key. It is possible to alter the image yet retain the same key, as the average is not always the best tool for characterizing data. Such a problem could be solved by exploring alternate ways of characterizing data. A non-linear mechanism might be more sensitive to small changes and is something that could be investigated.

One major flaw in our method is its inability to detect whether the alterations in the image are due to channel distortions and noise or actual tampering by an individual. Unfortunately, attempting to factor out transmission noise is a rather difficult task since it cannot always be modeled as a white noise effect. Sometimes the transmission noise is a function of the encoding scheme employed, and at other times it is a function of the channel itself. However, having a way to determine whether the "tampering" is the result of noise or a malicious attack would be useful. Ultimately though it is not that necessary. In order for the noise to be seen as tampering, it must be strong enough to start disrupting the image and from that point on it could be interpreted as an accidental attack.

Another potential problem is the "disgruntled employee" attack. If a disgruntled employee has access to the executable, then it is straightforward to make the executable always agree that the image received has not been tampered even if it has been. Similarly, the executable could be altered so that it gives consistently negative responses. Thus, there needs to be some way to prevent such attacks. One way to do so would be to introduce a random function that operates in conjunction with the executable, so that for example, a 5x5 local average key is not the only possibility.

Acknowledgments

I would like to thank Prof. Bede Liu and Min Wu for all of their help. They introduced me to the topic of watermarking and were always supportive. I would also like to thank Dr. Sharath Pankanti for his suggestions and advice. I would not have been able to write this paper without their assistance.

References

[1] A. K. Jain, L. Hong and S. Pankanti. "Biometric Identification", *Comm. ACM*, vol. 43, no. 2, pp. 91-98, Feb. 2000.

[2] S. Pankanti and M.Y. Yeung, "Verification Watermarks on Fingerprint Recognition and Retrieval", in *Proceedings of the SPIE/IS&T Electronic Imaging '99.*

[3] V.S. Nalwa. "Automatic On-line Signature Verification", in *Proceedings of the IEEE*, vol. 85, pp.215-239, Feb.1997.

[4] M. Wu, E. Tang and B. Liu. "Data Hiding in a Binary Image". To appear in *ICIP '00*.

[5] M. Riezenman. "Cellular Security: better, but foes still lurk", in *IEEE Spectrum*, vol. 37, pp. 39-42, 2000.

[6] R.G. van Schyndel, A.Z. Tirkel, and C.F. Osborne. "A Digital Watermark", in *Proceedings of the IEEE International Conference on Image Processing*, vol.II, pp. 86-90, 1994.

[7] I.J. Cox, J. Kilian, T. Leighton, and T. Shamoon. "Watermarking for Multimedia", NEC Research Institute Technical Report, 95-10, 1995.

[8] C. Hsu and J. Wu. "Hidden Signatures in Images". *IEEE ICIP III '96*, pp. 223-26.

[9] M. Wu, H. Yu, A. Gelman. "Multi-level Data Hiding for Digital Image and Video", *SPIE Photonics East '99*, Boston 1999.

[10] J. Fridrich, "A Hybrid Watermark for Tamper Detection in Digital Images", *ISSPA '99*. pp. 301-304.

[11] F. Minzer and G.W. Braudaway. "If One Watermark is Good, are More Better?", *IEEE '99*, pp. 2067 -2069.