Proceedings of the 6th Nordic Signal Processing Symposium - NORSIG 2004
June 9 - 11, 2004
Espoo, Finland

# ROBUST AUTHENTICATION OF THE JPEG 2000 BITSTREAM*

*Roland Norcen and Andreas Uhl*

Department of Scientific Computing, Salzburg University, AUSTRIA
e-mail: {rnorcen,uhl}@cosy.sbg.ac.at

## ABSTRACT

We discuss an authentication scheme for JPEG 2000 bitstreams in the sense of robust visual hashing. We show the robustness of our proposed method against JPEG 2000 and JPEG compression and discuss the sensitivity against malicious image modifications.

## 1. INTRODUCTION

The widespread availability of digital image and video data has opened a wide range of possibilities to manipulate these data. Compression algorithms change image and video data usually without leaving perceptual traces. Additionally, different image processing and image manipulation tools offer a variety of possibilities to alter image data without leaving traces which are recognizable by the human visual system.

In order to ensure the integrity and authenticity of digital visual data, algorithms have to be designed which consider the special properties of such data types. On the one hand, such an algorithm should be robust against compression and format conversion, since such operations are a very integral part of handling digital data. On the other hand, such an algorithm should be able to recognize a large amount of different intentional manipulations to such data.

Classical cryptographic tools to check for data integrity like the cryptographic hash functions MD-5 or SHA are designed to be strongly dependent on every single bit of the input data. While this property is important for a big class of digital data (for instance compressed text, executables, ...), classical hash functions cannot provide any form of robustness and are therefore not suited for typical multimedia data.

To account for these properties new techniques are required which do not assure the integrity of the digital representation of visual data but its visual appearance or content. In the area of multimedia security two types of approaches have been proposed so far: semi-fragile watermarking and robust multimedia hashes [1, 2, 4, 6, 7, 9]. Watermarking and robust visual hashing may be combined: hash values may be embedded into visual data using watermarking technologies, but in this case robust watermarking as employed for copyright protection is required.

A robust visual hashing scheme usually relies on a technique for feature extraction as the initial processing stage, often transformations like DCT or wavelet transform are used for this purpose. Subsequently, the features (a set of carefully selected transform coefficients) are further processed to increase robustness and/or reduce dimensionality. Two different approaches have been followed with respect to the final stage of the algorithms which has to produce the final hash value:

- The features are either directly converted into binary representation or fed into the decoder stage of error correcting codes or linear codes. This approach has the advantage that different hashvalues can be compared by evaluating the Hamming distance which serves as a measure of similarity in this case. Whereas it is desirable from the applications point of view to estimate the amount of difference between images by using those hash functions, this property severly threatens security and facilitates "gradient attacks" by iteratively adjusting hostile attacks to minimize a change in the hash value.
- A classical cryptographic hash function is applied to the extracted robust feature values. This approach guarantees security but the result is simply binary: image modification detected or not.

Authentication of the JPEG 2000 bitstream has been described in previous work. In [3] it is proposed to apply SHA-1 onto all packet data and to append the resluting hash value after the final termination marker to the JPEG 2000 bitstream. Contrasting to this approach, we focus onto robust authentication. This means that only certain parts of the bitstream are subject to authentication. The technical solution of how authentication can be applied to the entire codestream while it remains valid also for parts of it has been derived using Merkle hash trees [5] (and tested with MD-5 and RSA).

*This work has been partially supported by the Austrian Science Fund (project FWF-15170).

In this work we focus on the question which parts of the JPEG 2000 bitstream may serve as robust features in a sensible way. Section 2 shortly reviews JPEG 2000 and discusses application-related issues and basic considerations for robust JPEG 2000 authentication. Section 3 presents experimental results with respect to compression robustness and detection of malicious local image modifications.

## 2. AUTHENTICATING JPEG 2000

### 2.1. JPEG 2000 Basics

The JPEG 2000 [8] image coding standard uses the wavelet transform as energy compaction method. The major difference between previously proposed wavelet-based image compression algorithms such as EZW or SPIHT [8] is that JPEG 2000 operates on independent, non-overlapping blocks whose bit-planes are coded in several passes to create an embedded, scalable bitstream. JPEG 2000 may be operated in lossy and lossless mode (thereby using a reversible integer transform) and outperforms JPEG with respect to rate/distortion performance especially at lower bitrates.

The final JPEG2000 bitstream is organized as follows: The main header is followed by packets of data which are all preceded by a packet header. In each packet appear the codewords of the code-blocks that belong to the same image resolution (wavelet decomposition level) and layer (which roughly stand for successive quality levels). Depending on the arrangement of the packets, different progression orders may be specified. Among others, resolution and layer progression order are most important for grayscale images.

### 2.2. Robust JPEG 2000 Authentication

In the context of robust authentication it turns out to be difficult to insert the hash value directly into the codestream itself (e.g. after termination markers), since in any operation with involves decoding and recompression the original hash value would be lost (which should not automatically imply that the image content was changed significantly!). The only applications which do not destroy the hash value are purely bitstream oriented, like e.g. rate adaptation transcoding by simply dropping parts of the packet data. As a consequence, a possible solution to this dilemma would be to use a robust watermarking scheme to embed the hash value into the codestream, provided that the embedding does not change the features involved in computing the hash value. A different solution would be to signal the hash value in the context of a MPEG-7 or MPEG-21 description, separated but attached to the codestream. These questions are not further covered in this work, they are subject to further investigation.

In the following we restrict the attention to the assessment of different parts of the codestream with respect to their usefulness as robust feature values. Due to the embeddedness property of the bitstream, the perceptually more relevant bitstream parts are positioned at the very beginning of the file. Consequently, the bitstream is scanned from the very beginning to the end, and the data of each data packet - as they appear in the bitstream, excluding any header structures - are collected sequentially to be then used as visual feature values.

## 3. EXPERIMENTAL RESULTS

In this section we investigate if our proposed method is robust to JPEG 2000 recompression and JPEG compression on the one side, and sensitive to hostile local image alterations on the other side.

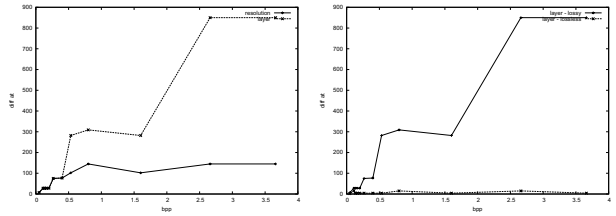
In our experiments we use classical 8bpp image data, including the well known Lena image at varying image dimensions ($512 \times 512$, $1024 \times 1024$, and $2048 \times 2048$ pixels), the plane image (see 2.a), and frame no. 17 from the surfside video sequence.

The experiments are conducted as follows: first, the feature values (i.e. packet data) are extracted from the JPEG 2000 codestream. Subsequently, the codestream is decoded and the image alteration is performed. Finally, the image is again JPEG 2000 encoded using the coding settings of the original codestream and the feature values are extracted and compared to the original ones.

The results which are presented in this section show the number of feature values (in bytes) required to detect an image modification (recall that packet data is used according to its appearance in the codestream). A value of - for instance - 42 means that the first 41 bytes of feature values (i.e. first 41 bytes of the codestream) are equal when comparing the modified image to the corresponding original codestream. The value itself can be easily interpreted: The higher the value, the more robust is the proposed method against the tested attack. In general, we want to see high values against JPEG 2000 and JPEG compression (robustness), but low values against local manipulations.

As a first step we test the JPEG 2000 recompression robustness by varying the coding options of the initial generation of the code stream (different parameters are used for feature extraction). These options include the JPEG 2000 standard parameter setting as well as coding in lossless mode, in resolution progression order, together with a varying wavelet-transform decomposition level. The JPEG 2000 compression (interpreted as image modification) is used in default

mode, i.e. layer progressive order, 5 decomposition levels (wlev5), and lossy coding.



(a) resolution progression (b) lossless coding

Figure 1: Different coding parameters used for feature extraction, lena512

Fig. 1 shows that if the parameters of JPEG 2000 compression and feature extraction match each other, the robustness against compression is very high. If they do not match, the robustness can be low and this is especially true if lossless coding is used for feature extraction (Fig. 1.b). The reason is that no quantization is used in lossless mode, therefore no robustness can be expected against compression. Since JPEG 2000 compression will often be used with default settings, we subsequently apply the feature extraction in lossy mode with layer progression order since we get maximal robustness in this case.

Tables 1 and 2 show the robustness of our proposed feature extraction mechanism against JPEG 2000 compression for different images. If the same coding options are used for feature extraction and for compressing the image, our method proves to be extremely robust against JPEG2000 compression (see table 2). This also means that JPEG 2000 encoding-decoding-encoding does not change the bitstream very much which was one of the design goals and is important for image editing applications.

| bpp | 4.5 | 2.66 | 0.8 | 0.4 | 0.2 | 0.133 | 0.05 |
|---|---|---|---|---|---|---|---|
| lena512 | 850 | 850 | 309 | 77 | 28 | 28 | 7 |
| lena1024 | 220 | 220 | 229 | 9 | 9 | 24 | 4 |
| lena2048 | 224 | 224 | 83 | 109 | 45 | 45 | 21 |
| plane512 | 293 | 293 | 64 | 41 | 33 | 33 | 5 |
| surf2048x1024 | 239 | 239 | 248 | 262 | 41 | 41 | 41 |

Table 1: Sensitivity against JPEG 2000 compression: wlev6 used for feature extraction, wlev5 used for JPEG2000 compression

Nevertheless, if different options are used, we can see a good robustness against moderate compression up to 1 bpp as well for all tested images (see Table 1).

Sensitivity against JPEG compression (see Table 3) is comparable to the sensitivity against JPEG 2000 compression in case the parameters do not match (Table 1) for better quality, at lower bitrates JPEG robustness is lower (which matches the poorer JPEG compression performance at low bitrates).

| bpp | 4.5 | 2.66 | 0.8 | 0.4 | 0.2 | 0.133 | 0.05 |
|---|---|---|---|---|---|---|---|
| lena512 | 4995 | 2561 | 309 | 547 | 309 | 187 | 187 |
| lena1024 | 4205 | 4205 | 1772 | 860 | 860 | 860 | 1772 |
| lena2048 | 1517 | 1517 | 6162 | 8119 | 7473 | 7910 | 4755 |
| plane512 | 1357 | 1357 | 466 | 817 | 466 | 460 | 233 |
| surf2048x1024 | 1795 | 1795 | 4964 | 4374 | 1054 | 1054 | 1054 |

Table 2: Sensitivity against JPEG2000 compression: identical coding options used for compression and feature extraction (wlev5)

| quality | 90 | 80 | 70 | 60 | 50 | 40 | 30 | 20 | 10 |
|---|---|---|---|---|---|---|---|---|---|
| lena512 | 42 | 77 | 42 | 42 | 67 | 42 | 1 | 28 | 15 |
| lena1024 | 24 | 296 | 24 | 24 | 9 | 9 | 24 | 4 | 4 |
| lena2048 | 232 | 109 | 125 | 31 | 65 | 45 | 21 | 22 | 3 |
| plane512 | 177 | 64 | 64 | 43 | 64 | 43 | 42 | 64 | 38 |
| surf2048x1024 | 306 | 41 | 41 | 41 | 41 | 27 | 7 | 56 | 7 |

Table 3: Sensitivity against JPEG compression: wlev5 used for feature extraction

The decomposition level used for the feature extraction can be used to influence the sensitivity against image alterations. This effect is shown in Table 4. We observe that a higher number of decomposition levels generally shows a higher sensitivity against image modifications including JPEG compression (see the left columns in table 4), and a smaller number decreases sensitivity against compression of this type - even against higher compression ratios (lower rows in the table, up to quality 50).

| wlev | 9 | 8 | 7 | 6 | 5 | 4 | 3 |
|---|---|---|---|---|---|---|---|
| quality 90 | 8 | 9 | 254 | 277 | 42 | 516 | 30 |
| quality 70 | 8 | 9 | 26 | 113 | 42 | 23 | 45 |
| quality 50 | 35 | 38 | 31 | 50 | 67 | 12 | 27 |
| quality 30 | 25 | 11 | 26 | 72 | 1 | 9 | 7 |
| quality 10 | 9 | 10 | 7 | 27 | 15 | 2 | 4 |

Table 4: JPEG compression (lena512): different wlev used for feature extraction

A wavelet decomposition level of 6 or 5 applied for feature generation seems to be well suited to result in satisfactory robustness against JPEG compression even at higher compression ratios.

Note that the presented extraction algorithm does not only have to be robust against compression, but also sensitive towards intentional image alterations. Here, a higher robustness against compression may mean that the algorithm is no longer able to be sensitive enough against other malicious image alterations. In order to investigate the sensitivity of our proposed scheme against intentional or malicious image alterations we have removed the US Air Force flag from the plane512 image (see Fig. 2.b).

In Table 5 we list the sensitivity results with respect to a chosen wavelet decomposition level. The wavelet decomposition level influences the ability of our algorithm to detect local image modifications significantly. Using a high value for wlev the local image modification is detected with a low number of feature

(a) plane original (b) plane - no flag

Figure 2: Testimage plane512 original and under attack.

values. At wlev 9, only 6 feature values are needed to detect the local attack.

| wlev | 9 | 8 | 7 | 6 | 5 | 4 | 3 |
|---|---|---|---|---|---|---|---|
| plane | 5 | 7 | 6 | 13 | 29 | 28 | 101 |

Table 5: Sensitivity against the removed flag.

As a consequence, there is the need for a compromise between the sensitivity against intentional image modifications on the one side, but robustness against JPEG2000 and JPEG compression on the other side. Regarding our results, we can say that a value for wlev of 5 or 6 seems to be best suited to be used for JPEG 2000 bitstream feature extraction. In this case, our method shows to be robust enough against compression up to a medium quality level, and the tested local attack can be detected with a rather low number of feature values. To give a concrete value based on these first results, we suggest to apply the hash function to the first 30 packet data bytes of the JPEG 2000 codestream to result in a robust authentication scheme.

## 4. CONCLUSION AND FUTURE WORK

We have shown that carefully selected parts of the JPEG 2000 bitstream can be employed as robust features. The presented method shows satisfying robustness against JPEG 2000 recompression and JPEG compression, and can detect intentional local image alterations. In future work we will focus on the sensitivity against Stirmark-type alterations and we will investigate possibilites how to insert the hash values into the codestream.

## 5. REFERENCES


[1] Jiri Fridrich. Visual hash for oblivious watermarking. In Ping Wah Wong and Edward J. Delp, editors, *Proceedings of IS&T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content II*, volume 3971, San Jose, CA, USA, January 2000.

[2] Jiri Fridrich and Miroslav Goljan. Robust hash functions for digital watermarking. In *Proceedings of the IEEE International Conference on Information Technology: Coding and Computing*, Las Vegas, NV, USA, March 2000.

[3] Raphaël Grosbois, Pierre Gerbelot, and Touradj Ebrahimi. Authentication and access control in the JPEG 2000 compressed domain. In A.G. Tescher, editor, *Applications of Digital Image Processing XXIV*, volume 4472 of *Proceedings of SPIE*, pages 95–104, San Diego, CA, USA, July 2001.

[4] T. Kalker, J. T. Oostveen, and J. Haitsma. Visual hashing of digital video: applications and techniques. In A.G. Tescher, editor, *Applications of Digital Image Processing XXIV*, volume 4472 of *Proceedings of SPIE*, San Diego, CA, USA, July 2001.

[5] Cheng Peng, Robert Deng, Yongdong Wu, and Weizhong Shao. A flexible and scalable authentication scheme for JPEG2000 codestreams. In *Proceedings of ACM Multimedia 2003*, pages 433–441, San Francisco, CA, USA, November 2003.

[6] R. Radhakrishnan, Z. Xiong, and N. D. Memom. Security of visual hash functions. In Ping Wah Wong and Edward J. Delp, editors, *Proceedings of SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents V*, volume 5020, Santa Clara, CA, USA, January 2003. SPIE.

[7] Champskud J. Skrepth and Andreas Uhl. Robust hash-functions for visual data: An experimental comparison. In F. J. Perales et al., editors, *Pattern Recognition and Image Analysis, Proceedings of IbPRIA 2003, the First Iberian Conference on Pattern Recognition and Image Analysis*, volume 2652 of *Lecture Notes on Computer Science*, pages 986–993, Puerto de Andratx, Mallorca, Spain, June 2003. Springer Verlag, Berlin, Germany.

[8] D. Taubman and M.W. Marcellin. *JPEG2000 — Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2002.

[9] Ramarathnam Venkatesan, S.-M. Koon, Mariusz H. Jakubowski, and Pierre Moulin. Robust image hashing. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'00)*, Vancouver, Canada, September 2000.