ROBUST VISUAL HASHING USING JPEG 2000

Roland Norcen and Andreas Uhl
Department of Scientific Computing, Salzburg University

Abstract Robust visual hash functions have been designed to ensure the data integrity of digital visual data. Such algorithms rely on an efficient scheme for robust visual feature extraction. We propose to use the wavelet-based JPEG2000 image compression algorithm for feature extraction. We discuss the sensitivity of our proposed method against different malicious data modifications including local image alterations and Stirmark attacks.

Keywords: Image authentication, robust feature extraction, JPEG 2000

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 perceptible traces. Beside, different image processing and image manipulation tools offer a variety of possibilities to alter image data without leaving traces which are recognizable to 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 MD5 or SHA-1 are designed to be strongly dependent on every single bit of the input data. This property is important for a big class of digital data (for instance compressed text, executables,...). Such classical hash functions are not suited for the class of 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 (Fridrich, 2000; Fridrich and Goljan, 2000; Kalker et al., 2001; Radhakrishnan et al., 2003; Skrepth and Uhl, 2003; Venkatesan et al., 2000).

Robust hash functions usually rely on a method for feature extraction to create a robust scheme for ensuring data integrity. Here, different algorithms have been proposed to extract a specific set of feature values from image or video data. The algorithms are designed to extract features which are sensitive to intentional alterations of the original data, but not sensitive to different standard compression algorithms like JPEG or JPEG2000.

The most efficient methods for feature extraction use transformation-based techniques. The DCT or the wavelet transform are two examples which can be employed in this case (Skrepth and Uhl, 2003).

In this work we discuss the possibilities how to use JPEG2000 for robust feature extraction. The basis for our method is a recently proposed algorithm (Norcen and Uhl, 2004) where an authentication scheme for JPEG2000 bitstreams is discussed, and its robustness regarding JPEG2000 and JPEG compression and recompression is shown. Here, we will show detailed results regarding the sensitivity towards local and global image alterations and we will discuss application scenarios how this approach can be used in real applications.

2. JPEG2000

The JPEG2000 (Taubman and Marcellin, 2002) image coding standard uses the wavelet transform as energy compaction method, and operates on independent, non-overlapping blocks whose bit-planes are coded in several passes to create an embedded, scalable bitstream.

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 (e.g., resolution and layer progression order).

2.1 Using the JPEG2000 bitstream for feature extraction

The JPEG2000 bitstream is analyzed for useful robust feature values. Therefore, 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.

Testing of the JPEG2000 coding options in Norcen and Uhl, 2004 showed the best set of coding parameters to be used for feature extraction: these options include the JPEG2000 standard parameter setting as well as coding in lossy mode in layer progression order, together with a varying wavelet-transform decomposition level.

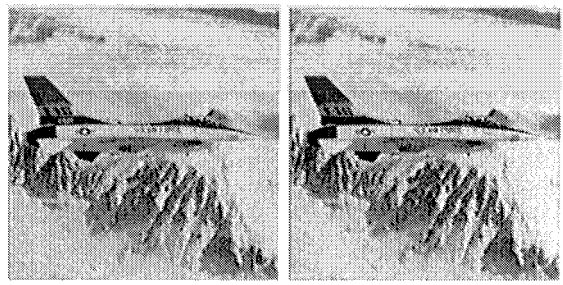
3. Experiments: Sensitivity Results

We use classical 8bpp image data in our experiments, including the well known lena image at varying image dimensions (512×512, 1024×1024, and 2048×2048 pixels), the houses (see 2.a), the plane (see 1.a), the graves image (see 3.a), the goldhill image (see 1.c), and frame no. 17 from the surfside video sequence (see 4.a). In the following we present detailed results regarding the sensitivity towards different local image alterations and global Stirmark modifications:

- local: different intentional image modifications:
  - plane: plane without call sign (see Figure 1.b)
  - graves: one grave removed (see Figure 3.b)
  - houses: text removed (see Figure 2.b)
  - goldhill: walking man removed (see Figure 1.d)
  - surfside frame: twisted head (see Figure 4.b)
- global: different Stirmark attacks (see `www.cl.cam.ac.uk/~mgk25/stirmark/`)

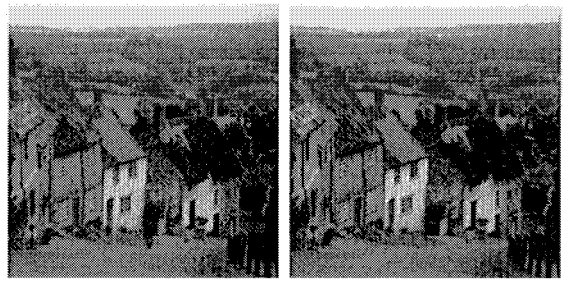
The experiments are conducted as follows: first, the feature values (i.e. packet data) are extracted from the JPEG 2000 codestream. Subsequently, the codestream is decoded and the image alteration is performed. Finally, the image is again JPEG 2000 encoded using the coding settings of the original codestream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

The results which are presented in the following show the number of feature values (in bytes) required to detect a global or local image modification. A value of – for instance – 42 means that the first 41 bytes



(a) plane original

(b) plane attacked – no call sign



(c) goldhill original

(d) goldhill attacked – without walking man

Figure 1. Local attacks.

of feature values are equal when comparing the computed features from the modified image to the feature values of the corresponding original image. 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 JPEG2000 and JPEG compression, but low values against all other tested attacks. Norcen and Uhl, 2004 showed that the feature extraction method is robust against moderate JPEG and JPEG2000 compression. In most cases, feature values of 50 or more were required for detecting JPEG and JPEG2000 compression ratios up to 1 or 0.8 bits per pixel. Here we want to detect all the described image alterations reliably. Therefore, we want to see significant lower feature values in all tests.

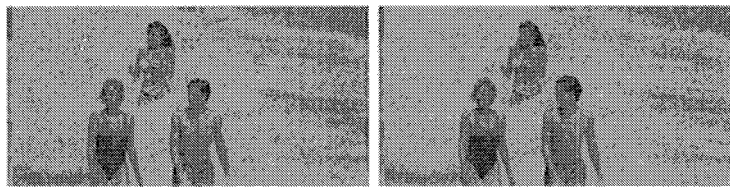
Table 1 lists the obtained results for the different local attacks with respect to a chosen wavelet decomposition level. The wavelet decom-

(a) houses original (b) houses attacked – without text

Figure 2. Local attacks.

(a) graves original (b) graves attacked – removed grave

Figure 3. Local attacks.



(a) surfside fr.17 original (b) surfside fr.17 – twisted head

Figure 4. Local attacks.

Table 1. Local attacks: different wlev used for feature extraction.

	wlev9	wlev8	wlev7	wlev6	wlev5	wlev4	wlev3
goldhill without man	7	7	28	44	29	48	155
houses without text	6	5	3	4	17	60	187
graves attacked	2	4	11	10	28	23	84
plane, no callsign	3	5	34	37	73	27	74
surfside, twisted head	6	17	7	20	2	68	412

position level obviously influences the ability of our algorithm to detect local image modifications. At a higher wlev parameter all local image modifications are detected with a low number of feature values. At wlev 9 for instance, only 7 feature values are needed to detect any of the tested local attacks. The modification of the graves image is detected with 2 feature values, in the plane image case only about 3 values are needed. At lower decomposition levels, more feature values are needed in general to detect the tested local image manipulations. At a wlev of 3, 412 feature values are needed to recognize the twisted head in the surfside frame, at wlev 4, only 68 are needed, and at the highest tested wlev, only about 6 are needed. Since the local changes are kept relatively small, the sensitivity regarding local image manipulations can be considered as high (depending on the wavelet decomposition level) – which of course is desired.

The Stirmark benchmark is used to rate the robustness and efficiency of various watermarking methods. Therefore, numerous image attacks are defined including rotation, scaling, median filtering, luminance modifications, gaussian filtering, sharpening, symmetric and asymmetric shearing, linear geometric transformations, random geometric distortions, and others. More details about the different attacks can be downloaded from the web page `www.cl.cam.ac.uk/~mgk25/stirmark/`, where the Stirmark testsetting is discussed at length. Our robust feature extraction method is tested against the standard Stirmark attacks, and due to the field of application our proposed method should be sensitive regarding all Stirmark attacks. In Table 2 a selection of the obtained results against global modifications is listed. Here we see the sensitivity against Stirmark attacks with parameter i, b, as well as global luminance modifications.

Again the results are delivered with respect to a chosen wlev for feature extraction, and only the results for the lena image at a resolution of 512×512 pixels are given. We can observe a high sensitivity against the presented global image alterations, except for a minimum change of the global luminance by a factor of 1, which shows a worse result.

Table 2. Different attacks/lena512: different wlev used for feature extraction.

	wlev9	wlev8	wlev7	wlev6	wlev5	wlev4	wlev3
stirmark i=1	1	3	6	1	5	1	1
stirmark i=2	1	6	7	2	6	1	1
stirmark b=1	1	6	6	2	3	1	1
stirmark b=2	1	4	5	12	1	1	1
luminance+1	1	4	7	12	36	9	3
luminance+2	1	1	7	2	12	9	3
luminance+3	1	1	6	1	6	5	3

Nevertheless, the sensitivity is high enough – as desired. Interestingly, a lower wlev parameter also shows a higher sensitivity against the Stirmark attacks with parameter i and parameter b. This effect can also be seen in other Stirmark attacked images. For this reason, a lower wlev could be preferred to be used for the feature extraction algorithm, since a lower wlev is also more robust against JPEG2000 and JPEG compression. However, all the local attacks presented in Table 1 could not be detected any longer when using such a low wlev parameter.

In Table 3 and Table 4 the results for the standard Stirmark testsetting are listed. Again, only results for the lena image at a resolution of 512×512 pixels are given with respect to a specific wlev. The first column of both tables clearly identifies the applied Stirmark attack and should be self-contained. Overall we can see that the sensitivity against all tested attacks is very high for a low and a high wlev value. For a wlev of 5 and 6, only the Gaussian filtering shows slightly higher feature values of about 36 and 23. Also a minor rotation and scale is slightly harder detectable. Here we need about 31 and 18 (wlev 5,6) feature values (see Table 4 first data row). The results for the other testimages are similar and therefore not listed here. In general, the sensitivity regarding Gaussian filtering as well as slight rotations and scalings is slightly inferior as compared to the other Stirmark tests. Regarding the graves image, these two test attacks are detected at a lower number of feature values, since the graves image is more sensitive to any image modification than the other tested images.

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 the robustness results in Norcen and Uhl, 2004, a wlev of about 6 or 5 seems to be best suited to be used for JPEG2000 bitstream feature extraction. In this case, we see a good sensitivity against local and global image

attacks, and robustness against JPEG2000 and JPEG compression up to moderate compression ratios.

4. Application Scenarios

Using parts of the JPEG2000 bitstream as robust visual features has important advantages, especially in the context of real world usability:

- Soft- and hardware to perform JPEG2000 compression will be readily available in large quantities in the near future which makes our proposed scheme a very attractive one (and also potentially cheap one).
- JPEG2000 Part 2 allows to use different types of wavelet transforms in addition to the Part 1 pyramidal scheme, in particular anisotropic decompositions and wavelet subband structures may be employed in addition to freedom in filter choice. This facilitates to add key-dependency to the hashing scheme by concealing the exact type of wavelet decomposition in use, which would create a robust message authentication code (MAC) for visual data. This could significantly improve the security against attacks (compare Meixner and Uhl, 2004).
- Most robust feature extraction algorithms require a final conversion stage to transform the computed features into binary representation. This is not necessary since JPEG2000 is of course given in binary representation.

We get two scenarios where our method can be applied in a straightforward manner: first, our method can be applied to any raw digital image data, via computing the JPEG2000 bitstream and then the JPEG2000 feature values. Second, any JPEG2000 bitstream can be used itself as starting point. In this case, the considered bitstream is the original data which should be protected, and the features are extracted directly from the investigated JPEG2000 bitstream. This scenario is useful, where some image capturing device directly produces JPEG2000 coded data instead of raw uncompressed data (i.e. JPEG200 compression implemented in hardware, no raw data saved).

After having extracted the feature values out of the JPEG2000 bitstream, three strategies may be followed:

- The extracted features are fed into the decoder stage of error correcting codes or linear codes to reduce the number of hash bits and to increase robustness. This approach has the advantage that different hash strings can be compared by evaluating the Hamming distance which serves as a measure of similarity in this case.

Table 3. Standard stirmark testsetting, lena512: different wlev used for feature extraction.

	wlev9	wlev8	wlev7	wlev6	wlev5	wlev4	wlev3
17_row_5_col_removed	4	4	3	1	2	1	1
1_row_1_col_removed	5	6	26	7	12	5	7
1_row_5_col_removed	6	4	15	2	12	2	1
3x3_median_filter	3	1	3	7	1	4	13
5_row_17_col_removed	4	4	9	1	5	2	1
5_row_1_col_removed	4	4	7	7	5	5	1
5x5_median_filter	3	1	3	4	1	12	13
7x7_median_filter	3	1	3	4	3	4	8
9x9_median_filter	3	1	3	4	3	4	13
Gaussian_filtering_3_3	1	5	7	23	36	5	23
Sharpening_3_3	1	4	7	2	15	9	3
cropping_1	4	4	6	1	5	1	1
cropping_10	1	3	1	1	1	1	1
cropping_15	1	1	1	1	1	1	1
cropping_2	2	4	6	1	5	1	1
cropping_20	2	1	1	1	1	1	1
cropping_25	3	1	1	1	1	1	1
cropping_5	1	4	1	1	1	1	1
cropping_50	1	2	1	1	1	1	1
cropping_75	1	1	1	1	1	1	1
flip	1	1	1	1	1	1	1
linear_1.007_0.010_0.010_1.012	1	2	2	1	1	2	1
linear_1.010_0.013_0.009_1.011	1	2	2	1	1	2	1
linear_1.013_0.008_0.011_1.008	1	2	2	1	1	2	1
ratio_x_0.80_y_1.00	3	1	3	1	1	1	1
ratio_x_0.90_y_1.00	3	1	3	1	1	1	1
ratio_x_1.00_y_0.80	1	1	2	1	1	1	1
ratio_x_1.00_y_0.90	1	4	3	1	2	1	1
ratio_x_1.00_y_1.10	1	1	3	1	2	1	2
ratio_x_1.00_y_1.20	1	1	1	1	2	1	1
ratio_x_1.10_y_1.00	1	2	1	2	1	1	1
ratio_x_1.20_y_1.00	1	2	2	2	1	1	1
rotation_-0.25	4	4	6	2	6	1	1
rotation_-0.50	4	4	6	1	5	1	1
rotation_-0.75	4	4	6	1	4	1	1
rotation_-1.00	4	4	6	1	1	1	1
rotation_-2.00	2	4	4	1	1	1	1
rotation_0.25	4	4	12	2	12	1	1
rotation_0.50	4	4	6	2	12	1	1
rotation_0.75	4	4	6	2	12	1	1
rotation_1.00	2	4	6	2	6	1	1
rotation_10.00	2	3	1	1	1	1	1
rotation_15.00	2	1	1	1	1	1	1
rotation_2.00	2	3	6	1	6	1	1
rotation_30.00	1	1	1	1	1	1	1
rotation_45.00	1	1	1	1	1	1	1
rotation_5.00	2	3	1	1	1	1	1
rotation_90.00	1	1	1	1	1	1	1

Table 4. Standard stirmark testsetting, lena512: different wlev used for feature extraction.

	wlev9	wlev8	wlev7	wlev6	wlev5	wlev4	wlev3
rotation_scale_-0.25	4	4	6	18	31	4	11
rotation_scale_-0.50	4	4	6	16	6	4	5
rotation_scale_-0.75	4	4	6	1	6	4	1
rotation_scale_-1.00	4	4	6	1	1	1	1
rotation_scale_-2.00	1	4	4	1	1	2	2
rotation_scale_0.25	6	4	6	2	3	1	5
rotation_scale_0.50	6	4	6	2	3	1	1
rotation_scale_0.75	6	4	6	2	3	1	1
rotation_scale_1.00	3	4	6	2	3	1	1
rotation_scale_10.00	2	3	1	1	1	1	1
rotation_scale_15.00	2	3	1	1	1	1	1
rotation_scale_2.00	2	4	6	2	1	1	1
rotation_scale_30.00	1	1	1	1	1	1	1
rotation_scale_45.00	1	1	1	1	1	1	1
rotation_scale_5.00	2	3	1	1	1	1	1
rotation_scale_90.00	1	1	1	1	1	1	1
scale_0.50	4	1	1	1	1	1	1
scale_0.75	2	1	1	1	1	1	1
scale_0.90	2	4	3	1	1	1	1
scale_1.10	1	2	1	1	1	1	1
scale_1.50	1	1	1	1	1	1	1
scale_2.00	1	1	1	1	1	1	1
shearing_x_0.00_y_1.00	5	4	11	1	3	5	1
shearing_x_0.00_y_5.00	3	4	3	1	1	1	1
shearing_x_1.00_y_0.00	4	5	7	2	12	1	1
shearing_x_1.00_y_1.00	5	6	7	1	5	1	1
shearing_x_5.00_y_0.00	1	3	2	2	4	1	2
shearing_x_5.00_y_5.00	1	3	2	1	4	1	1

Whereas it is desirable from the point of view of the applications to estimate the amount of difference between images by using those hash functions, this property severely threatens security and facilitates "gradient attacks" by iteratively adjusting hostile attacks to minimize a change in the hash value.

- A classical cryptographic hash function (like MD5 or SHA-1) is applied to the feature data to result in an overall robust but cryptographically secure robust visual hash procedure. The possibility to measure the amount of difference between two hash strings is lost in this case, however, gradient attacks and other security flaws are avoided.

- The extracted feature values are used as hash strings as they are without any further processing. The obvious disadvantages in terms of the higher amount of hash bits and lower security against attacks is compensated by the possibility to localize and approximately reconstruct detected image alterations since the hash string contains data extracted from a low bitrate compressed version of the original image.

In the latter case, with the available feature value data (consisting of JPEG2000 packet body data), and the corresponding packet headers which need to be generated and inserted into the codestream, the original image can be reconstructed up to the point the codeblock data is available in the packet bodies. A packet header indicates, among other information, which codeblocks are included in the following packet body, whereas the body contains the codeblocks of compressed data itself. Without the packet header, a reconstruction of the corresponding packet body is not possible in general. Therefore, these packet headers need to be inserted.

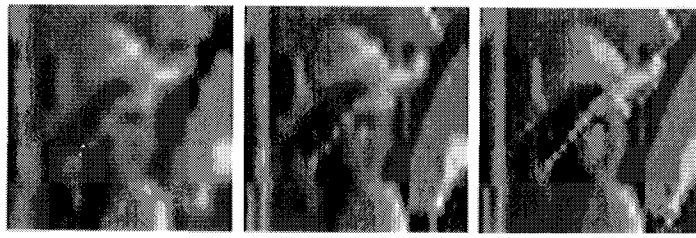
In Figures 5 and 6 we visualize the approximations of the original images using feature value data of the lena and the graves image only. In each case, the first 512, 1024, and 2048 bits of feature values are used.

Since the given number of feature value bits which are used for the visual reconstruction include packet body data only, the overall number of bits used for reconstruction – including the needed packet header data – must be somewhat bigger. Table 5 shows the number of bits which are required for the corresponding images. The first column gives the number of feature bits used, and the entries in the table show the overall number of bits which are needed for the visual reconstruction. We see that a considerable number of "extra" bits are needed. These "extra bits" stem from the corresponding packet headers and are needed

to reconstruct the image data up to the point where codeblock packet body data is given in the features.

Table 5. Signature bits (including packet header data).

	lena512	graves512	plane512
512 bits	552	552	552
1024 bits	1144	1136	1136
2048 bits	2224	2208	2224



(a) 512 bits (b) 1024 bits (c) 2048 bits

Figure 5. Reconstruction of lena.

(a) 512 bits (b) 1024 bits (c) 2048 bits

Figure 6. Reconstruction of graves.

The number of feature bits used have been chosen in a way to demonstrate a possible application where the hash string could be signed using a digital signature algorithm like ElGamal or RSA. In this context, using a 512 feature bits signature already could help to localize and approximately reconstruct severely manipulated regions in the image, whereas a

2048 feature bits signature allows to gain information about some details as well.

5. Conclusion

The JPEG2000 algorithm can be employed to extract robust features from an image. The presented method has shown to be robust against moderate JPEG2000 and JPEG compression. In this work we showed that the method is also very sensitive regarding global and local image alterations including Stirmark attacks and different intentional local image modifications. Application scenarios for our approach are discussed and show this method to be of interest for practical employment.

References

Fridrich, Jiri (2000). Visual hash for oblivious watermarking. In Wong, Ping Wah and Delp, Edward J., 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.

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

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

Meixner, Albert and Uhl, Andreas (2004). Analysis of a wavelet-based robust hash algorithm. In Delp, Edward J. and Wong, Ping W., editors, *Security, Steganography, and Watermarking of Multimedia Contents VI*, volume 5306 of *Proceedings of SPIE*, San Jose, CA, USA. SPIE. To appear.

Norcen, R. and Uhl, A. (2004). Robust authentication of the JPEG2000 bitstream. In *CD-ROM Proceedings of the 6th IEEE Nordic Signal Processing Symposium (NORSIG 2004)*, Espoo, Finland. IEEE Norway Section.

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

Skrepth, Champskud J. and Uhl, Andreas (2003). Robust hash-functions for visual data: An experimental comparison. In Perales, F. J. 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. Springer Verlag, Berlin, Germany.

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

Venkatesan, Ramarathnam, Koon, S.-M., Jakubowski, Mariusz H., and Moulin, Pierre (2000). Robust image hashing. Vancouver, Canada.