BLOCK CONVERGENCE IN REPEATED TRANSFORM CODING: JPEG-100 FORENSICS, CARBON DATING, AND TAMPER DETECTION

ShiYue Lai and Rainer Böhme

University of Münster, Dept. of Information Systems, Leonardo-Campus 3, 48149 Münster, Germany

ABSTRACT

Repeated rounding of sample blocks in alternating domains creates complex convergence paths. We study convergence and block stability for JPEG images compressed with quality factor 100 and derive methods to detect such compression in grayscale bitmap images, to estimate the number of recompressions, to identify the DCT implementation used for compression, and to uncover local tampering if image parts have been compressed with JPEG-100 at least once.

1. INTRODUCTION

Detecting prior JPEG compressions and estimating the quantization tables (QTs) from bitmap images are key techniques in digital image forensics. Prior art can be classified into three approaches. *First*, approaches confined to the spatial domain measure the strength of blocking artifacts. These methods can be turned to JPEG detectors by defining thresholds to distinguish between pre-compressed and never-compressed images. The common principle of Fan et al.'s [1] and Muijs and Kirenko's [2] methods is to compare pixel value differences within blocks and across blocks boundaries. Pan et al. [3, 4] refine these methods with edge detectors by looking for edge orientations of 0 or 90 degrees. *Second*, approaches working in the DCT domain can detect JPEG compression and estimate its QT. Fu et al. [5] apply digit analysis to DCT coefficients and, stipulating that Benford's law holds for this data generating process, derive test statistics to detect deviations from that law caused by (pre-)compression with (a set of) a candidate QT(s). Luo et al. [6] study how rounding errors affect the distribution of DCT coefficients. Their theory enables to detect double JPEG-compression even in small (parts of) images. The observation that quantization and rounding leaves periodic traces in the DCT histogram can be exploited by applying spectral theory on the joint effect of quantization and rounding. The analysis of the characteristic function (i.e., DFT of the DCT histogram) allows precise estimates of individual quantization factors in the primary or secondary QT [7, 8]. Pevný and Fridrich [9] use machine learning to detect JPEG double-compression and estimate the primary QT for the special case that the suspect image is given as JPEG and possibly contains steganography. The *third* approach is to use cross-domain techniques. Bovik et al. [10] and Park et al. [11] measure blocking artifacts in the spatial domain with frequency-domain features. Fridrich et al.'s [12] compatibility test checks for each block of an image if it could have plausibly been generated from a decompression operation within a range of quantization factors. Böhme [13] assigns the most likely among a set of candidate QTs based on the residuals of recompression attempts. Lewis and Kuhn [14] propose exact recompression by inverting specific implementations of the decompression function using interval algebra. Their method can determine all possible QTs of color images by eliminating impossible quantization factors. Closest to this work, Huang et al. [15] count the number of changed coefficients after recompression to detect double-compression with the same QT.

This wealth of literature indicates a mature field. Yet we are not aware of any method to detect JPEG compression with quality factor 100 (henceforth JPEG-100), where the QT contains only ones and quantization coincides with rounding. Moreover, few methods generalize to image histories with more than two compression–decompression cycles. We close these gaps with a novel approach that exploits convergence properties of transform-coded blocks.

The remainder of this paper is structured as follows. Section 2 describes our observations of block convergence, Sect. 3 presents forensic tools based on the analysis of block convergence, Sect. 4 reports quantitative results to demonstrate their effectiveness, and the final Sect. 5 concludes.

2. JPEG-100 BLOCK CONVERGENCE

Let $\boldsymbol{x}^0$ denote the intensity matrix of a grayscale image, where superscript 0 indicates that it has never been compressed. Then, a history of repeated JPEG-100 compression and decompression operations is modeled by this recursion:

$$\boldsymbol{x}^{t+1} = \mathrm{tr}\left([\mathrm{IDCT}([\mathrm{DCT}(\boldsymbol{x}^t)])]\right),\ t \geq 0, \qquad (1)$$

where $[.]$ denotes rounding, $\mathrm{tr}(.)$ truncates to the value range, and DCT and IDCT are 8×8 block-wise DCT and inverse DCT transformations. We call a block *stable* (after t iterations) if its values in $\boldsymbol{x}^{t+1}$ equal its values in $\boldsymbol{x}^t$. A key observation is that repeated rounding in spatial and transformed domains causes a complex convergence path where the number of iterations until a block is stable follows a distribution that is largely independent of the image content. To understand this effect and learn how it can be used for JPEG-100 forensics, we start with simple situations.

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

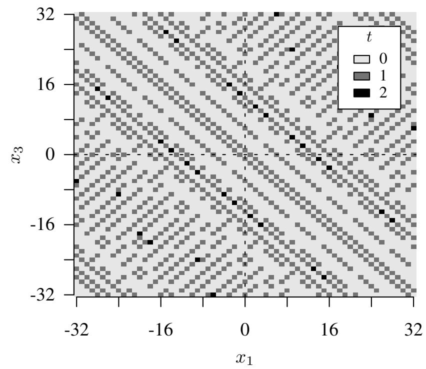

Fig. 1: Map of convergence time t for triples $(x_1, 0, x_3)$

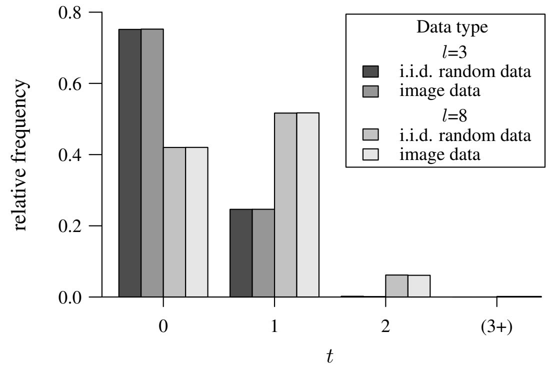

Fig. 2: Empirical PMF $p(t)$ for 1-D blocks: comparison of i.i.d. random uniform data and samples of natural images.

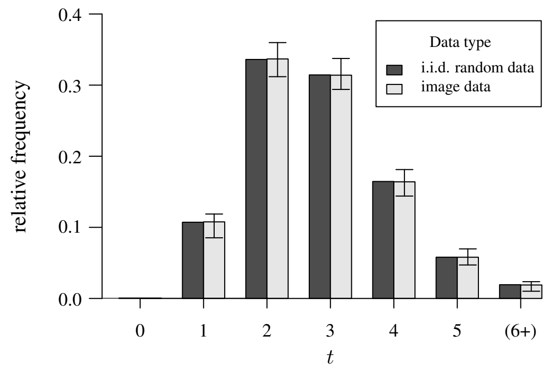

Fig. 3: Empirical PMF $p(t)$ for 2-D 8×8 blocks: comparison of i.i.d. random data uniform data and natural image blocks. The intervals show the full value range over all tested images.

2.1. Observations for 1-D blocks

We begin with an exhaustive study of 1-D blocks of the form $(x_1, 0, x_3)$, with $x_1, x_3 \in \{-32, \ldots, 32\}$. Because any constant integer offset only adds to the value of the DC coefficient in the transformed domain and does not affect rounding, it is sufficient to plot the time t until convergence in a 2-D plane of Fig. 1 and the picture generalizes to arbitrary triples of integers. All tested blocks become stable within $t \leq 2$ rounds. Moreover, except for linear slopes, the distribution of convergence times appears rather chaotic and there is no indication that macroscopic properties of the block substantially predict its time to convergence. This indicates that the convergence time of any given block in natural images can be modeled as an independent realization of a discrete distribution $p(t)$, $t \geq 0$. Moreover, the empirical ratio of stable blocks in never-compressed natural images approaches $p(0)$ as the number of blocks grows, but is largely independent of the image content.

To test this conjecture, we compare the empirical PMF $p(t)$ of 3.8 million random i.i.d. uniform triples (x_1, x_2, x_3) in the range $[0, 255]^3$ to the PMF of 3.8 million samples drawn from 800 different never-compressed grayscale images. As evidenced in Fig. 2, for block length $l = 3$, there is no visible difference between random and image data. The same holds if we expand the block length to $l = 8$, though fewer blocks are stable at $t = 0$ and some probability mass moves to $t > 2$.

It is important to note that the results are obtained with flat blocks excluded. We call a block *flat* if all x_i have the same value. All AC coefficients of flat blocks are zero, hence they are stable at $t = 0$, independent of the dimension. Since many images contain flat blocks and their amount varies widely between images, without excluding them, the PMF of a specific image would not necessarily match the PMF over all images.

2.2. Extension to 2-D blocks

Now we extend our study to 2-D blocks sized 8×8, like in JPEG-100 compression. After removing the flat blocks, we plot in Fig. 3 the PMF $p(t)$ for i.i.d. random data and for all blocks of 800 natural images. (All following figures are based on the same image data.) The close match indicates that our conjecture generalizes to realistic blocks. Moreover, as the range of possible t increases in 2-D, and few blocks are stable at $t = 0$, we may infer t from the ratio of stable blocks r. Let b_{flat}, b_{stable}, and b_{total} denote the number of flat, stable, and the total number of blocks, respectively. Then r is given by:

$$r = (b_{\text{stable}} - b_{\text{flat}}) \,/\, (b_{\text{total}} - b_{\text{flat}}) \ . \qquad (2)$$

2.3. Influence of the DCT implementation

The data presented so far was generated using what we call textbook DCT, i.e., a full matrix multiplication using transform coefficients from the definition of the DCT. In practice, faster FDCT algorithms employ a divide-and-conquer strategy to minimize the number of multiplications. FDCT methods either factorize the DCT similar to the FFT, that is breaking an N-point DFT into two $N/2$-point DFTs, or directly break an N-point DCT into two $N/2$-point DCTs. Although mathematically equivalent, because of finite-precision arithmetics, different DCT implementations have different numerical properties, which may affect the block convergence.

`libjpeg` is a widely used library providing JPEG decoding and encoding functions alongside various utilities for

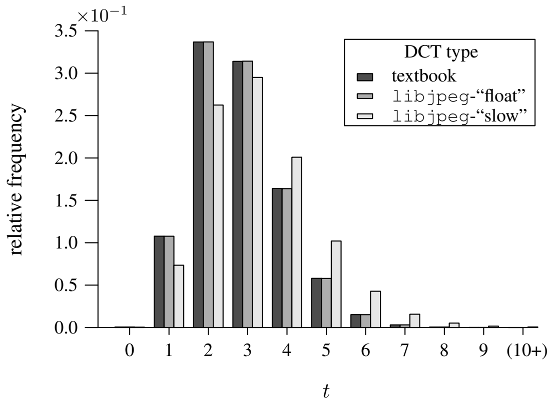

Fig. 4: Influence of the DCT implementation on $p(t)$

handling JPEG images [16]. It supports three different implementation of FDCT, called "fast", "float" [17], "slow" [18][1]. Note that "float", "slow" , "fast" are just labels. All variants are fast DCTs using different algorithms and number formats. To avoid confusion, we put these labels in quotation marks.

Fig. 4 shows that the PMFs of textbook and "float" DCTs are very similar, but convergence takes a little longer on average for the "slow" type, `libjpeg`'s default method. The "fast" type (not plotted) is completely different. Its integer arithmetic generates nearly no stable blocks for $t < 6$. As it produces visibly bad quality, this method is barely used in practice and thus excluded from the analysis in this paper.

Blocks which are stable under one DCT implementation may be unstable under another. Given an image $\boldsymbol{x}^t$ that has been compressed t times with one DCT implementation (denoted by $\mathrm{DCT}_{\mathrm{pre}}$), we iterate through a set of candidate implementations ($\mathrm{DCT}_i, i = 1, 2, \dots$) and calculate r_i. We expect that the maximum ratio of stable blocks is most likely reached if $\mathrm{DCT}_i = \mathrm{DCT}_{\mathrm{pre}}$. Fig. 5 confirms this effect. Observe that the gap between the maximum $r_{\hat{\mathrm{pre}}}$ and other r_i grows with t.

3. FORENSICS USING BLOCK CONVERGENCE

Next we show how our observations enable simple yet effective forensic tools for images with possible JPEG-100 history.

3.1. JPEG-100 carbon dating with known $\mathrm{DCT}_{\mathrm{pre}}$

Suppose a forensic analyst knows the fixed DCT implementation used to compress the image under analysis with JPEG quality factor 100. Then the following JPEG "carbon dating" method can estimate the number of recompressions $\hat{t}$, which can be an indication of how often the image has been touched by image processing software. Given a bitmap image $\boldsymbol{x}$, calculate the ratio of stable blocks r using Eq. 2. Compare the result to a series of critical values obtained from the range of

[1]The "float" and "fast" type DCTs are both borrowed from Arai, Agui, and Nakajima's algorithms for scaled DCT. Though original written in Japanese, the two algorithms are both described in English in [17].

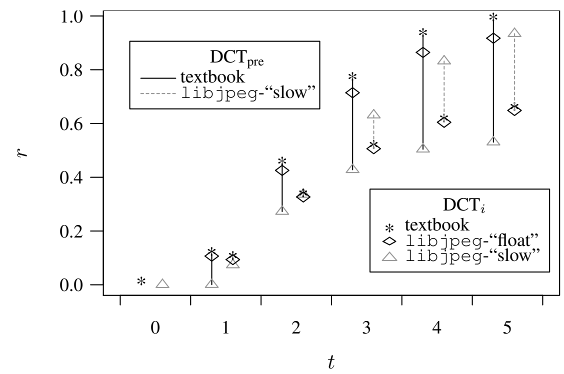

Fig. 5: Block stabilization indicates DCT implementation

r for random i.i.d. uniform data using the known $\mathrm{DCT}_{\mathrm{pre}}$ and different candidate t. If r is smaller than the lower end of the interval associated with $t = 1$, consider the image as never-compressed. If r is closer to one than the highest upper end below one, then it is impossible to estimate t exactly, but the image must have been recompressed at least $t_{\max}$ times, where $t_{\max} = 4$ for textbook and "float", and $t_{\max} = 5$ for "slow".

3.2. Identification of the DCT implementation

If the analyst does not know the exact $\mathrm{DCT}_{\mathrm{pre}}$, but has knowledge about and access to all candidate implementations DCT_i, she can try to identify the implementation by finding $\arg\max_i r_i$. This method can be used to select the appropriate set of critical values in the carbon dating procedure, or in completely different contexts, where knowledge about the DCT implementation reveals relevant facts about an image's provenance (e.g., in forensic camera model identification).

3.3. Uncovering local tampering

We illustrate by example how block convergence analysis can be used in local tamper detection by spotting inconsistencies in the density of stable blocks. Fig. 6 shows a forgery in which all stable blocks are marked as white frames. Note the exceptionally low density in the tampered region. In this example, the original has been compressed once with JPEG-100 before processing. A similar difference in the density of stable blocks (albeit on a different level) is observable if the complete forgery is stored or transmitted as JPEG-100.

4. EXPERIMENTAL VALIDATION

We report the results of two large experiments to benchmark the proposed methods. Both experiments were run on three different image databases, 1600 raw images captured with a single Minolta DiMAGE A1 camera, downscaled to 640×480 pixels and converted to grayscale, 800 never-compressed images sized 3000×2000 from the Dresden Image Database [19], and 800 never-compressed images sized 3600×2400

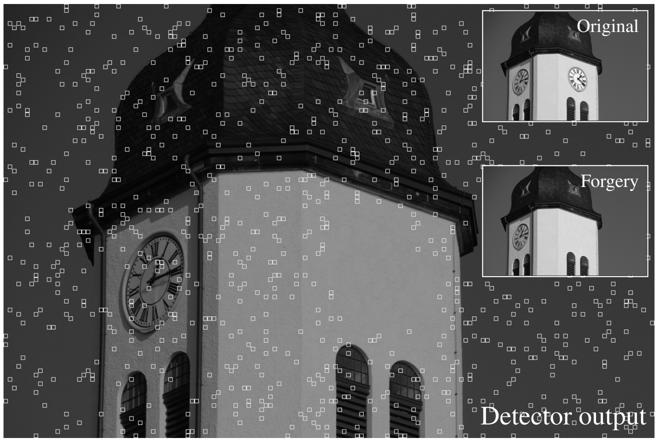

Fig. 6: Inconsistent distribution of stable blocks uncovers local tampering. One clock has been removed from a JPEG-100 precompressed image ($t = 1$) using a copy stencil tool.

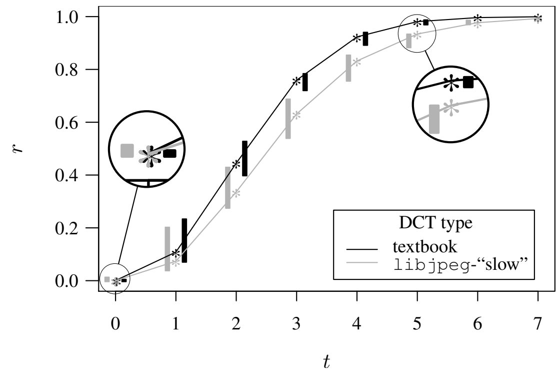

Fig. 7: Cumulative ratio of stable blocks r as a function of the number of JPEG-100 recompressions t for two DCT implementations. The intervals show the range of r in 800 natural images and serve as critical values for carbon dating. The curve for textbook DCT also applies to `libjpeg`-“float”.

from the BOSSbase [20]. The first 1600 images were randomly and equally divided into two groups. 800 of them were used for the observation of convergence effect in Sect. 2 and the other 800 images were reserved for validation.

4.1. Results for JPEG-100 carbon dating

Our experiments show that if we know the DCT implementation and use the critical values plotted in Fig. 7, then t can be correctly estimated for *all* images in three different image databases. There was not a single misclassification, so we refrain from printing confusion tables. To assess the importance of excluding flat blocks, we repeated the analysis with flat blocks included (using adjusted critical values). We observe 3.6 % misclassifications for “slow”, and 3.1 % for textbook and “float”. Relatively more errors happen for small t and all errors overestimate the actual number of recompressions.

4.2. Results for DCT implementation identification

Table 1 reports the detection results and misclassifications for 800 images from the Minolta database for $t \in \{1, 2\}$. If $t > 3$, then the detection becomes perfect as the difference between the maximum and the alternative candidates increases (cf. Fig. 5). In particular for $t = 1$ and “slow”, we face misclassifications in the order of 50 %. Note that this does not directly translate into error rates for carbon dating if this identification method is used to select critical values. This is because for small t, where the detection is error-prone, the critical intervals are mutually exclusive between different t even if we join the ranges for all tested DCT implementations (cf. Fig. 7) . In other words, the accuracy of carbon dating depends on the exact knowledge of the DCT implementation only in ranges of t where the identification works perfectly.

Table 1: Confusion table of DCT identification performance

	True DCT implementation					
	$t = 1$			$t = 2$		
Detector output	text-book	“float”	“slow”	text-book	“float”	“slow”
textbook	604	183	382	800	0	68
“float”	196	617	396	0	800	86
“slow”	0	0	22	0	0	646

Note: Perfect identification for $t \geq 3$.

5. CONCLUSIONS AND OUTLOOK

We have introduced block convergence analysis as a new approach that closes gaps in the existing literature on JPEG forensics. Our methods allow a forensic investigator to reliably detect prior JPEG compression with quality factor 100, to estimate the number of recompressions (which may indicate how often an image has been opened and resaved in image processing software), to identify the DCT implementation used for compression (which may reveal information about acquisition devices or processing software), and to identify local tampering (filtering, splicing, resampling) if image parts have been compressed with JPEG-100 at least once. We note that JPEG-100 is very relevant in practice if forgers save intermediate versions of tampered images with the highest possible quality, but refrain from switching to a lossless format.

Directions for future work include a theory for the distribution of convergence times, experiments with more DCT implementations, the generalization of the carbon dating and DCT detection to JPEG quality factors below 100, and the extension to color images, where subsampling might affect the convergence behavior of multiple adjacent blocks.

6. REFERENCES

[1] Z. Fan and R.D. Queiroz, "Identication of bitmap compression history: JPEG detection and quantizer estimation," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 230–235, February 2003.

[2] R. Muijs and I. Kirenko, "A no-reference blocking artifact measure for adaptive video processing," in *Proceedings of the 13th European Singal Processing Conference (EUSIPCO)*, Antalya, Turkey, 2005.

[3] F. Pan, X. Lin, S. Rahardia, E.P. Ong, and W.S. Lin, "Measuring blocking artifacts using edge direction information," in *IEEE International Conference on Multimedia and Expo (ICME)*, June 2004, pp. 1491–1494.

[4] F. Pan, X. Lin, S. Rahardia, E.P. Ong, and W.S. Lin, "Using edge direction information for measuring blocking artifacts of images," *Multidimensional Systems and Signal Processing*, vol. 18, no. 4, pp. 297–308, December 2007.

[5] Dongdong Fu, YunQing Shi, and Wei Su, "A generalized Benford's law for JPEG coefficients and its applications in image forensics," in *SPIE Conference on Security, Steganography, and Watermarking of Multimedia Contents*, San Jose, CA, January 2007, vol. 6505.

[6] WeiQi Luo, JiWu Huang, and GuoPing Qiu, "JPEG error analysis and its applicaitons to digital image forensics," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 480–491, September 2010.

[7] ShuiMing Ye, QiBin Sun, and Ee-Chien Chang, "Detecting digital forgeries by measuring inconsistencies of blocking artifact," in *IEEE International Conference on Multimedia and Expo (ICME)*, July 2007, pp. 12–15.

[8] Thomas Gloe, "Demystifying histograms of multi-quantized DCT coefficients," in *IEEE International Conference on Multimedia and EXPO (ICME)*, 2011, pp. 1–6.

[9] Tomáš Pevný and Jessica Fridrich, "Detection of double-compression in JPEG images for applications in steganography," *IEEE Tansactions on Information Forensics and Security*, vol. 3, no. 2, pp. 247–258, June 2008.

[10] Alan C. Bovik and ShiZhong Liu, "DCT-domain blind measurement of blocking artifacts in DCT-coded images," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2001, vol. 3, pp. 1725–1728.

[11] Chun-Su Park, Jun-Hyung Kim, and Sung-Jea Ko, "Fast blind measurement of blocking artifacts in both pixel and DCT domains," *Journal of Mathmatical Imaging and Vision*, vol. 28, no. 3, pp. 279–284, 2007.

[12] Fridich Jessica, Miroslav Goljan, and Rui Du, "Steganalysis based on JPEG compatibility," in *SPIE Conferece on Multimedia Systems and Applications IV*, Andrew G. Tescher, Bhaskaran Vasudev, and Bove Jr., Eds., Denver, CO, 2001, vol. 4518, pp. 275–280.

[13] Rainer Böhme, "Weighted stego-image steganalysis for JPEG covers," in *Information Hiding*, K. Solanki, K. Sullivan, and U. Madhow, Eds., Berlin Heidelberg, 2008, vol. 5284 of *LNCS*, pp. 178–194, Springer-Verlag.

[14] Andrew B. Lewis and Markus G. Kuhn, "Exact JPEG recompression," in *SPIE Conference on Electronic Imaging, Security and Forensics of Multimedia XII*, San Jose, CA, January 2010, vol. 7543, p. 7543V.

[15] Fang Jun Huang, JiWu Huang, and YunQing Shi, "Detecting double JPEG compression with the same quantization matrix," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 848–856, December 2010.

[16] Independent JPEG Group, "libjpeg," `http://libjpeg.sourceforge.net`, 1998.

[17] William B. Pennebaker and Joan L. Mitchell, *JPEG: Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1993.

[18] Christoph Loeffler, Adriaan Lieenberg, and George S. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1989, vol. 3, pp. 988–991.

[19] Thomas Gloe and Rainer Böhme, "The Dresden Image Database for benchmarking digital image forensics," *Journal of Digital Forensic Practice*, vol. 3, pp. 150–159, 2010.

[20] P. Bas, T. Filler, and T. Pevný, "Break our steganographic system," in *Information Hiding*, T. Filler, T. Pevný, S. Craver, and A. Ker, Eds., Berlin Heidelberg, 2011, vol. 6958 of *LNCS*, pp. 59–70, Springer-Verlag.