A new approach to high-capacity annotation watermarking based on digital fountain codes

Paweł Korus · Jarosław Białas · Andrzej Dziech

Abstract Annotation watermarking is a technique that allows to associate content descriptions with digital images in a persistent and format independent manner. It is commonly used in medical applications and, hence, existing schemes have been designed to meet rigorous watermark transparency requirements. As a result, the effective capacity of such schemes is severely limited. In this paper, we present a new approach to annotation watermarking. We adopt the fountain coding paradigm and design a convenient watermark communication architecture which resembles a traditional packet network. Our approach allows for straightforward incorporation of content adaptivity, robustness against cropping and support for multiple data streams. In our study, we focus on high-capacity annotations and we assume different requirements with respect to the fidelity of the watermarked images. Our scheme is robust against lossy JPEG compression and cropping. This paper describes the principles of the proposed approach and presents the results of it's experimental evaluation.

Keywords Digital watermarking · Annotation watermarking · Watermark coding · Multiple watermarking · Digital fountain codes

1 Introduction

Efficient searching and filtering of digital images requires easily accessible textual descriptions of their content. Annotations or tags are commonly used for this purpose. These labels can be associated either with the whole images or with their selected fragments. Annotations often carry ground truth data for evaluation of object detection and recognition algorithms. Thus, carefully annotated image sets

P. Korus (✉) · J. Białas · A. Dziech
AGH University of Science and Technology, al. Mickiewicza 30, 30-059 Kraków, Poland
e-mail: korus@kt.agh.edu.pl

are highly valued in computer vision communities. In practical systems, annotations are often used to store the results of automatic object recognition or classification.

One of the problems with traditional annotations is that they are stored separately from the images. Among the most common choices are relational databases or auxiliary description files. Additional meta-data can also stored in dedicated containers, e.g., eXtensible Metadata Platform (XMP) which can either use format-specific storage or the auxiliary meta-data files. To the best of our knowledge, XMP does not allow for attaching the descriptions to specific fragments of the media files. An additional problem is that these descriptions are often lost when transmitting the images without an explicit care for the annotations or during prospective media format conversions.

Annotation watermarking is a technique that allows to embed these descriptions into digital images in a persistent and format-independent manner [3]. The annotations are tied to the images by means of imperceptible modifications of their appearance. The watermark usually needs to be robust against common image processing operations, like brightness adjustments, lossy compression, scaling or cropping.

Embedding annotations via watermarks is particularly common in medical applications [6, 16]. The typical annotations include the identifiers of both the doctor and the patient as well as the most relevant extract of the medical history of the latter. The payload of such schemes is limited as one of the most important factors is the negligible quality impact of the embedded information. In certain cases, it is required that the watermarking process should be fully reversible.

In the considered application, the annotations are associated with selected polygons on the image and a cropped fragment of the original image is expected to provide a valid description of that particular region. An annotation watermarking scheme with similar application in mind has been presented in [13]. However, the effective capacity of 0.012 bpp is not sufficient for our needs. This limitation is common for all of existing annotation watermarking systems and stems from the typically assumed requirement of minimal possible quality impact. A summary of existing schemes will be presented in Section 4.

In this paper we present a new approach to annotation watermarking with emphasis on the user data payload. Our requirements with respect to the image quality are different from the commonly assumed ones. In our application, it is not required to deliver complete perceptual transparency of the watermark. The lack of visible artifacts is fully sufficient. Depending on the number of annotations and on the resolution of the image, our scheme allows to embed up to 0.2 bpp of effective user data payload. Assuming an acceptable distortion is approx. 35 dB, the scheme is robust against lossy JPEG compression with nearly error-free watermark recovery up to the quality level of 70. Below that level, it is still possible to recover the watermark provided that the descriptions are short.

Our approach adopts the fountain coding paradigm [12] for the purpose of encoding the watermark's payload. The designed communication architecture resembles a traditional packet network and uses a dedicated mechanisms for multiplexing data streams from multiple annotations. The implemented decoder does not need to know the capacity assignment and recovers the appropriate selection channel on its own. Thus, our approach allows for straightforward incorporation of both robustness against cropping and content adaptivity. The descriptions of cropped

image fragments are recovered with automatic translation of the necessary polygon coordinates.

This paper is an extended version of a conference paper [10]. The remaining part of this paper is organized as follows. Section 2 describes the principles of the proposed approach. The results of experimental evaluation are shown in Section 3. Conclusions, supplemented with a comparison with existing schemes are presented Section 4.

2 Fundamental concepts of the proposed scheme

The proposed annotation scheme uses two independent watermarks. The first one allows for synchronization with the original block grid in case of cropping. The second one carries the payload of the annotations and the necessary headers. The general idea of the considered scheme is shown in Fig. 1. The original and the watermarked images are denoted as x and x^*, respectively. w represents the auxiliary watermark and $W_n^{(i)}$ the nth symbol of ith annotation's watermark.

The first step of the encoder is to divide the image into blocks. Successive steps of the algorithm require different block sizes, which imposes a three-level block hierarchy, described in more detail in Section 2.2.
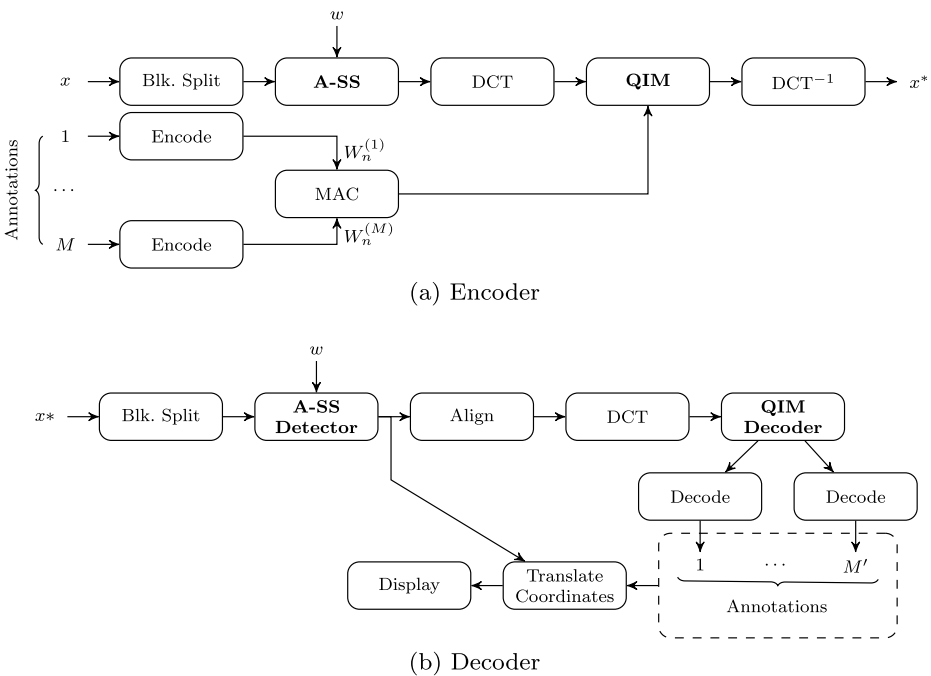

Fig. 1 Operation of the annotation encoder and decoder; x—cover image, $w \in \{-1, 1\}^{64 \times 64}$—synchronization watermark, additive spread spectrum, x^*—watermarked image

The first of the watermarks is embedded in the spatial domain using the additive spread-spectrum technique [3]. A correlation detector implemented in the decoder recovers the shift between the original and the current block grid. The details of this synchronization procedure are presented in Section 2.2.

The next step is to perform block-based forward Discrete Cosine Transform (DCT). This domain allows for straightforward selection of frequencies least affected by the prospective JPEG compression. Due to high capacity requirements, we have used the Distortion Compensation Quantization Index Modulation (DC-QIM) technique for embedding the main watermark [1, 4]. Each coefficient eligible for carrying the watermark is modified according to:

$$\hat{x}_{i,j}^* = (1 - \gamma) \cdot \text{sign}\left(\hat{x}_{i,j}\right) \cdot \Delta \cdot Q_m\left(\frac{|\hat{x}_{i,j}|}{\Delta}\right) + \gamma \cdot \hat{x}_{i,j} \tag{1}$$

where $\hat{x}_{i,j}$ is the cover image coefficient, $\hat{x}_{i,j}^*$ is the watermarked coefficient, Δ is the quantization step and γ is the distortion compensation parameter. $Q_m(\cdot)$ is a quantizer for message bit m:

$$Q_m(x) = \begin{cases} 2 \cdot \lfloor \frac{x}{2} + 0.5 \rfloor & \text{if } m = 0, \\ 2 \cdot \lfloor \frac{x}{2} \rfloor + 1 & \text{if } m = 1 \end{cases}$$

For the purpose of generating the main watermark, the payload of each annotation is encoded by a fountain code [12] and supplemented with the necessary headers. The Medium Access Control (MAC) module assigns the available capacity and multiplexes the data streams from multiple annotations. In the last step, the modified spectrum is transformed back to the spatial domain using inverse DCT.

The decoder begins by detecting the spatial domain synchronization watermark. Based on the detected translation, it aligns the image to match the original block division grid. Then, the image is transformed to the DCT domain and the QIM watermark is recovered. After stripping and analyzing the headers with necessary configuration data, the detector begins to decode the streams of all identified annotations. The last step is to perform coordinate translation of the polygons' vertices.

The operation of all of the relevant steps of the algorithm will be described in detail in dedicated sections.

2.1 Annotation transport architecture

The principle of the proposed annotation watermarking scheme is to deliver a layered architecture analogous to traditional packet networks. The payload of each annotation is divided into constant length symbols, which are encoded in order to introduce the necessary redundancy. For this purpose, we adopt the fountain coding paradigm [12]. Its fundamental assumption is that successful decoding is possible from arbitrary fragments of the symbol stream. The only requirements is that the decoder needs to receive a certain portion of the transmitted symbols. For an ideal code, the necessary portion would be exactly as long as the original message. Practical codes, however, introduce additional overhead.

Due to different properties of existing digital fountain codes, the proposed scheme allows for selection of the most appropriate one based on the needs of each particular message. We consider two basic codes: the random linear fountain (RLF) [12] and

the LT code [11]. The latter can be configured to operate in different variants by choosing the most appropriate degree distribution and the preferred decoding algorithm. Thus, one can configure the system to achieve a good balance between the low overhead of the random linear fountain and the low decoding cost of the LT code.

A fountain code produces output symbols by calculating linear combinations of random input symbols. The number of combined input symbols is referred to as a *degree* of the output symbol. This degree can be chosen either uniformly, like in RLF, or according to a selected degree distribution, like in LT codes. The degree distribution has crucial impact on the properties of the code. The presented system implements three degree distributions: the Robust Solution Distribution (RSD) [11], the Revised RSD (RRSD) [2] and the optimized degree distribution (OPTD) [8].

The RSD is the distribution proposed in the original paper on LT codes, it balances the average degree so that the decoding process would proceed without interruptions and would not require an excessive number of operations per symbol. The main problem with this distribution is a large overhead for short messages. This limitation is mitigated by incorporation of the OPTD. The last of the considered distributions, the RRSD, allows for further improvement of the decoding performance at the cost of even higher overhead.

Another way to balance the overhead vs. decoding complexity trade-off is to select the appropriate decoding algorithm. In the described system, it can be chosen to be Gaussian elimination (GE), Belief Propagation (BP) or hybrid. The latter begins with BP and uses GE when BP can no longer proceed.

A fountain code is capable of delivering a limitless symbol stream. In the considered system, the output stream length is constant and stems from the size of the image. One output symbols is generated for every macro block of the image. The MAC module decides which of them are actually going to be used. In practice, the unnecessary output symbols do not need to be calculated. The fountain code, however, needs to explicitly generate null symbols for the sake of proper encoder-decoder synchronization.

The choice of the symbol length stems from experimental evaluation of the impact of JPEG compression on each of the transform coefficients. The described system uses 60-bit symbols with additional 16-bit hash for error detection. This choice can be easily adapted to the requirements of a particular application and it does not affect the performance of the utilized fountain codes. The described transmission architecture is shown in Fig. 2.

2.2 Block hierarchy and grid synchronization

Each of the steps of the proposed scheme uses a different block size. The designed three-layer block hierarchy is shown in Fig. 3. The lowest-level division is based on the 8×8 px grid used by JPEG. Thus, it allows for direct assessment of the impact of lossy compression on individual frequencies of the spectrum.

The capacity of an individual lowest-layer block is insufficient for the discussed application. Thus, for the purpose of embedding the watermark symbols W_n, we group 4 lowest-layer blocks into 16×16 px *macro blocks*, which are capable of carrying one symbol of the watermark payload. Each 8×8 px block carries 19 bits of the watermark payload. The information is embedded into first 19 coefficients
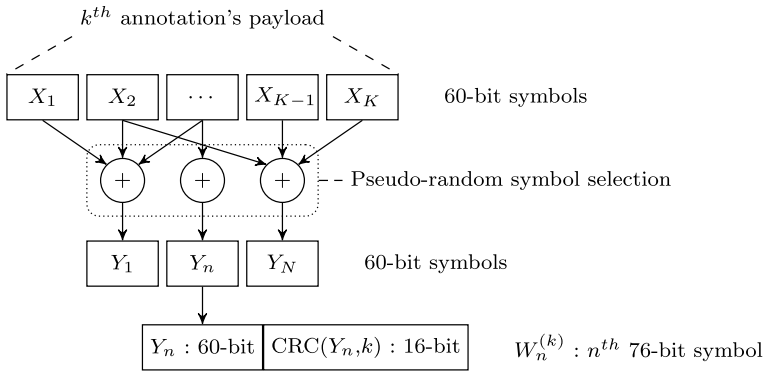
Fig. 2 Encoding and encapsulation of the payload of a single annotation

of the DCT spectrum in the zig-zag order. Due to excessive quality impact, the DC coefficient is not eligible for watermark embedding. First four coefficients from each block are used to embed the high-priority symbol hash.

The highest layer of the hierarchy groups 16 macro blocks into 64 × 64 px *synchronization blocks*. The function of synchronization blocks is twofold. Firstly, they impose a strict organization of the macro blocks. Two first macro blocks are reserved for the necessary system headers and are referred to as the *scheme configuration block* and the *stream configuration block*. The former carries fundamental properties of the scheme, i.e., the number of embedded annotation streams and the dimensions of the original image, which are necessary for synchronization of the fountain decoders and for translation of the annotations' coordinates in case of cropping. This information is repeated in every such block in the whole image.

The second reserved block defines the properties of the embedded streams, i.e., their lengths and fountain code configuration. Due to capacity limitations, each stream configuration block describes the parameters of up to three annotation
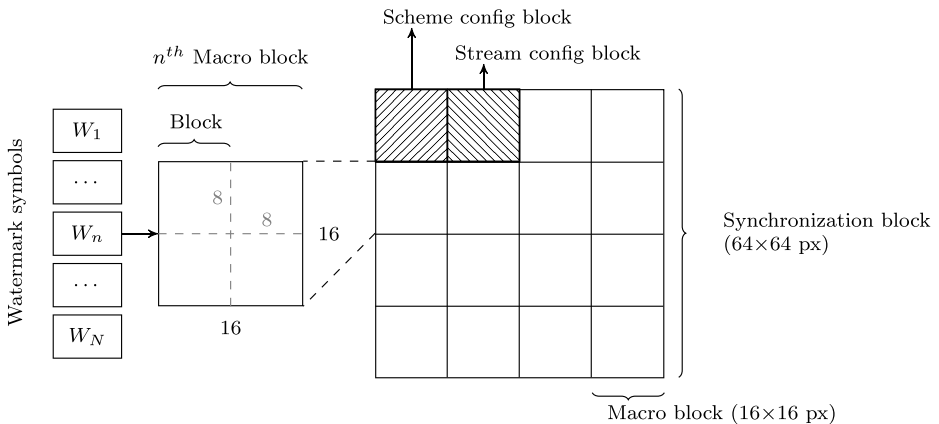

Fig. 3 The considered three-layer block hierarchy

streams. By spatial multiplexing, it is possible to describe the necessary configuration data for all of the embedded streams.

The seconds use for the synchronization blocks is that they are used for embedding the auxiliary spread spectrum watermark. A uniform bipolar pseudo-random pattern $w \in \{-1, 1\}^{64 \times 64}$ is tiled to match the image size. We use the additive spread spectrum technique for embedding in the spatial domain (2).

$$x_{i,j}^* = x_{i,j} + \alpha w_{i \bmod 64, j \bmod 64} \tag{2}$$

This auxiliary watermark allows for rapid resynchronization with the original blocking grid [9]. The detector calculates the correlation of an average synchronization block $\overline{x}$ with the known watermark pattern w. The location of the watermark detection peak corresponds to the grid misalignment vector. This principal is illustrated in Fig. 4 which shows an exemplary misalignment between the original and the cropping-inflicted block division grid and the corresponding detector response.

For the sake of computational efficiency, the decoder calculates the correlation in the Fourier domain. The magnitude of the spectrum is discarded to increase the detection performance [9]. Hence, the decision is based solely on the angle between the inspected vectors, which essentially corresponds to the correlation coefficient detector [3]. The correlation matrix C is obtained by coefficient-wise multiplication of the image and watermark spectra:

$$C = f^{-1} \left(\Phi \left(f \left(\overline{x} \right) \right) \cdot \Phi \left(f(w) \right) \right) \tag{3}$$

where $f(x)$ is the Fast Fourier Transform and $\Phi(x)$ is a magnitude discarding function:

$$\Phi(x) = \begin{cases} \frac{x}{|x|} & \text{if } x \neq 0, \\ 1 & \text{otherwise} \end{cases}$$

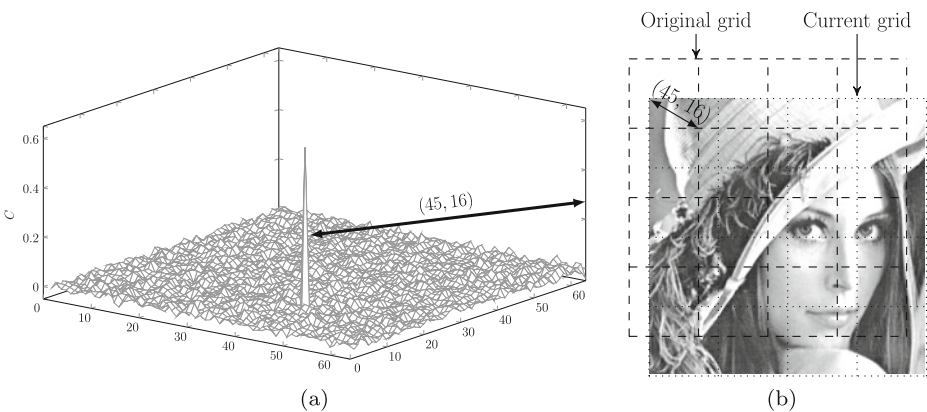This detector is equivalent to the Symmetric Phase Only Matched Filtering (SPOMF).

Fig. 4 An exemplary watermark detection result. The peak in (**a**) corresponds to the detected grid misalignment from (**b**)

2.3 Medium Access Control

The MAC module is responsible for multiplexing the data streams from multiple annotations, i.e., it assigns the macro-blocks to the concurring streams. It needs to take into account two principal factors. Firstly, the requirement that the description should be recoverable from a cropped version of the relevant image fragment. Secondly, the necessary overheads in order to determine the amount of actually needed symbols.

The determined assignment is not communicated to the decoder, which by validating the hash values of the embedded symbols, is capable of restoring the macro-block to data stream mapping. This map is supplemented incrementally, i.e., with each successive annotation, the decoder needs to check a quickly decreasing image area.

The operation of the MAC begins with initial assignment of the macro-blocks to the data streams by using the minimal-area bounding shape for each of the defined polygons on the image. The shape it determined with macro-block accuracy. The resulting assignment is highly susceptible to conflicts, even if there are no overlapping polygons involved in the process. Hence, the next step is usually the conflict resolution procedure. For this purpose the MAC builds an implicit hierarchy of the defined annotations, based on their mutual location and the overlap area. In case the conflict is related to a parent-child relation, the conflict is resolved in favor of the child polygon. The parent will get compensation from it's surroundings. In case the conflicting regions are both children of a common parent, the decoder takes into account the necessary overhead for the associated descriptions.

When all of the conflicts have been resolved, the MAC estimates the necessary overhead for all of the defined annotations. If there is any shortage with respect to this criterion, the problematic region is expanded iteratively with a dilation-like operation. Expansion of child regions is allowed only within the parent regions. The whole supplementation process is performed iteratively, starting from the leafs of the built implicit hierarchy.

In the final step, the MAC repeats the dilation-like expansion of the top-most regions to fill the remaining unassigned area of the image. This process is illustrated in Fig. 5: (a) shows the defined polygons overlaid on a tinted cover image, (b) shows the initial assignment of the available capacity, (c) shows the result of conflict resolution process and (d) shows the final assignment after the final top-level expansion.
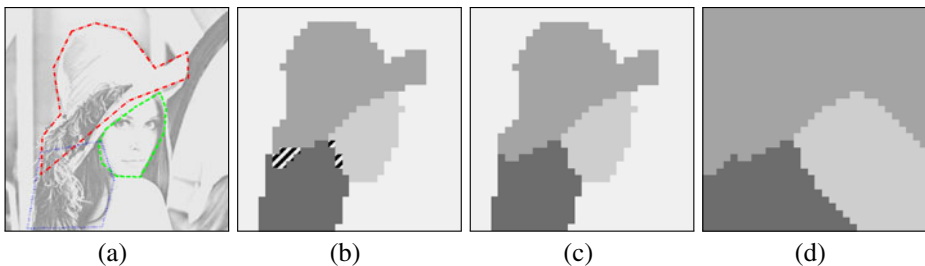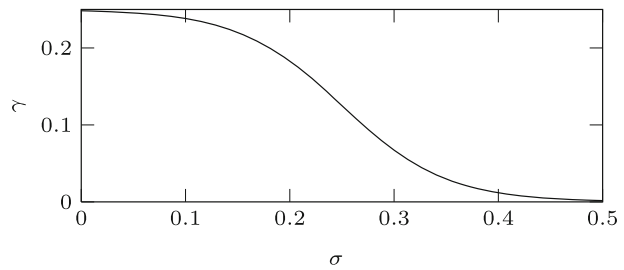

| (a) | (b) | (c) | (d) |

Fig. 5 Successive steps of medium access control: **a** defined polygons, **b** initial assignment with marked conflicts, **c** assignment after conflict resolution, **d** final assignment

Fig. 6 The mapping of the
block standard deviation to
distortion compensation

2.4 Content adaptation

One of the advantages of the proposed approach is straightforward support for content adaptation. This functionality is implemented solely in the encoder, which needs to assess the necessary embedding parameters in order to guarantee successful message decoding on the receiver side. Analogous to the capacity assignment by the MAC module, the decoder does not need to be aware of any content adaptation mechanisms.

The adaptation is carried out by proper adjustments of the distortion compensation parameter γ, which can be used to mitigate the embedding distortion for solid image areas. A Human Vision System (HVS) model can be used to decide which macro blocks are more suitable for information embedding. This is a challenging problem in digital information hiding, as the decoder needs to estimate the original HVS model to recover the selection channel [3]. This often leads to sub-optimal perceptual models, which tend to be robust against prospective content modifications. There also exist techniques for *non-shared selection channel* communication which select the best embedding variant from the ones that will be recovered as the desired message in the decoder. These techniques are more frequently adopted in digital steganography and the selection criterion usually stems from certain payload detectability estimates [5].

In this study, we use a simple model which adapts the distortion compensation γ for a whole macro-block based on the standard deviation of it's normalized pixel values σ:

$$\gamma = \frac{1}{4}\left(1 - \left(1 + e^{-20(\sigma - 0.25)}\right)^{-1}\right)$$

This mapping function is shown in Fig. 6. The compensation is higher for solid image blocks, where the embedding artifacts become visible more easily. The embedding is not eliminated completely, in order not to create visible boundaries between the eligible and non-eligible image blocks.

3 Experimental evaluation of the main performance measures

In this study, we focus on three main performance aspects of the proposed scheme. Firstly, we evaluate the quality impact of both of the watermarks. The assessment takes into account both uniform and distortion compensated (Section 2.4) embedding. Secondly, we validate the robustness against lossy compression and cropping. We address this issue both from the grid synchronization and data retrieval

perspectives. The last aspect is the capacity of the proposed scheme, i.e., we experimentally evaluate the overhead of the implemented fountain codes and assess the achievable effective payload of user data.

In our tests, we have used a test set of 90 images, partially originating from the commonly used USC database and partially being selected photographs from the Flickr photo sharing service. The test set contains all commonly seen types of content, from drawings through photo-realistic computer graphics to traditional photographs.

3.1 Assessment of the quality impact

The considered scheme uses two independent watermarks: a spatial domain auxiliary watermark and the main DCT domain watermark. The former is embedded using the additive spread spectrum technique (2) with constant embedding strength $\alpha = 1$ for all of the performed experiments. This value is both the lowest possible as well as fully sufficient for the application at hand. The average distortion introduced by this watermark is 48 dB in terms of the Peak Signal to Noise Ratio (PSNR).
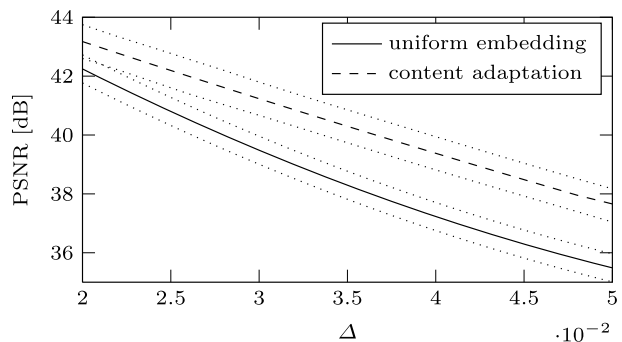
The main watermark is embedded into all macro-blocks of the image. This simplifies the quality assessment as the only factor that influences the introduced distortion is the quantization step Δ, i.e., the strength of the watermark. Figure 7 shows the average distortion vs. Δ along with 95% prediction intervals for both of the embedded watermarks.

The embedded watermark introduces a noise-like pattern to the image and is not disturbing for a human observer. Figure 8 shows an exemplary fragment of a watermarked image with different embedding strengths. Figure 8a–d illustrate uniform embedding with no distortion compensation. Figure 8e shows the result of adopting the described model of adaptive compensation. The achieved quality improvement is approximately 2 dB and is reflected in visibly better fidelity of the solid image areas.

3.2 Robustness against cropping and lossy compression

We consider the robustness of the proposed scheme against cropping and lossy JPEG compression. Cropping causes grid misalignment and the decoder needs to be able to resynchronize prior to watermark recovery. The average success rate of $99.5 \pm 0.2\%$ with 95% confidence has been calculated from 7,200 independent replications of

Fig. 7 The average joint quality impact for both of the embedded watermarks with 95% prediction intervals
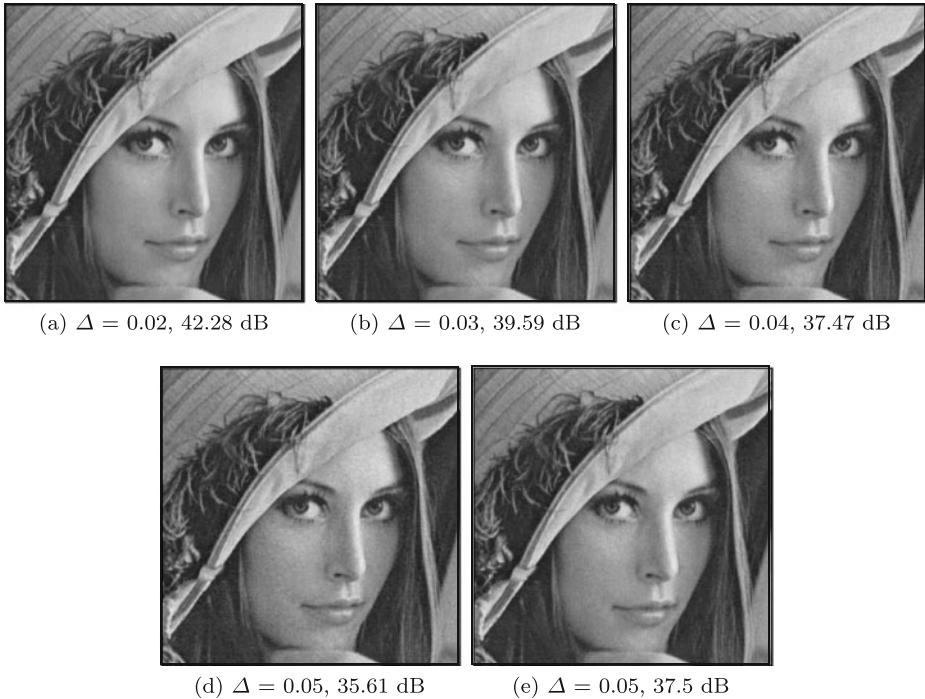
(a) $\Delta = 0.02$, 42.28 dB   (b) $\Delta = 0.03$, 39.59 dB   (c) $\Delta = 0.04$, 37.47 dB

(d) $\Delta = 0.05$, 35.61 dB   (e) $\Delta = 0.05$, 37.5 dB

Fig. 8 Resulting watermarked images for different embedding strengths Δ

the experiment on the described test set of 90 images. The test scenario included all possible grid misalignment vectors with step 2 and in combination with lossy compression from the range [80; 100] with step 5. Since the lowest-level blocks are 8×8 px, we take into account only this range of possible grid misalignment.

In addition to the misalignment compensation, proper synchronization requires also successful recovery of the scheme configuration and embedding parameters. The quantization step Δ is estimated from the distribution of the DCT coefficients. Then the decoder recovers the content of the scheme configuration blocks. The success rate of this complete synchronization process is shown in Fig. 9a. The plot shows the average success rate from various random cropping patterns.

The most important performance criterion is the watermark symbol recovery rate η. In this paper, we consider only flawlessly recovered symbols. Implementation of error correction mechanisms is expected to provide a considerable improvement. Due to a limited length of this paper, this issue is out of scope of this study. The average recovery rate vs. the quantization step Δ and the JPEG quality setting is shown in Fig. 9b. For the highest considered embedding strength, which corresponds to $\approx$35 dB in terms of the PSNR and uniform embedding, the watermark can be fully recovered ($\eta \approx 100\%$) for JPEG compression up to the quality level of 75. In the range 65–75 there remains a fair amount of recoverable symbols and short messages are still likely to be successfully decoded.

The symbol recovery rate slightly deteriorates when adopting the described content adaptation mechanism. Figure 9c shows an exemplary slice of the symbol
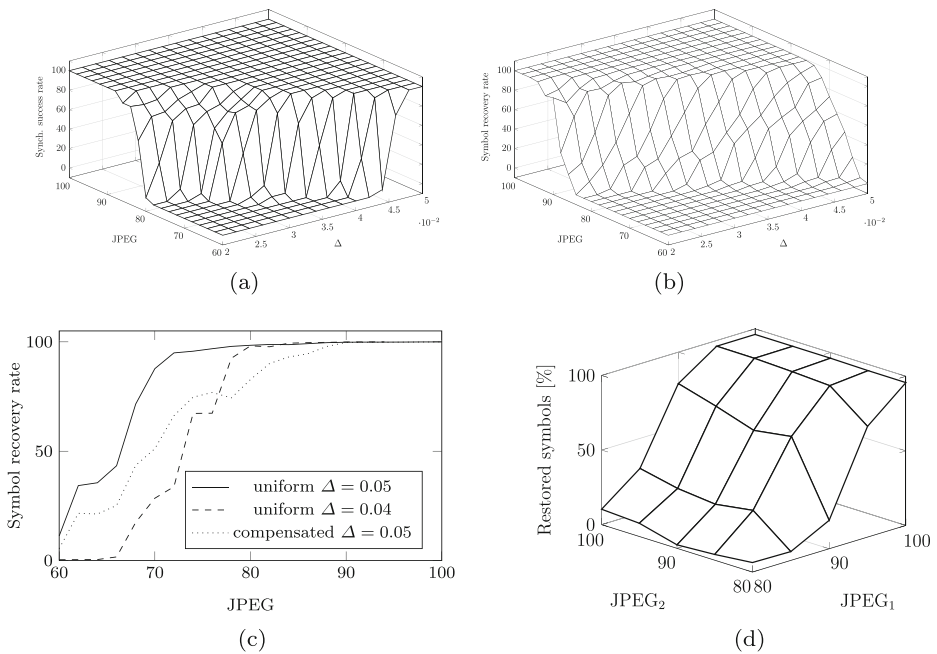
Fig. 9 a Synchronization recovery rate for different quantization steps and lossy compression settings, **b** symbol recovery rate for different quantization steps and lossy compression settings, **c** comparison of robustness for uniform and distortion compensated embedding **d** symbol recovery rate for cropping and double JPEG compression for $\Delta = 0.05$

recovery rate experiment for the highest considered embedding strength $\Delta = 0.05$. We find this example the most informative as lower strengths hardly benefit visually from content adaptation. The uniform embedding with $\Delta = 0.04$ is shown for reference only as the PSNR for this case is comparable to the compensated embedding.

When comparing the symbol recovery rate degradation point, the performance of all of the variants is similar and the degradation starts around the JPEG quality level of 90. The degradation steepness is the most precipitous for the variant with content adaptation. However, after some point, the robustness of the adaptive variants remains between the two uniform ones.

The performed evaluation of symbol recovery rate also shows that the developed synchronization procedure successfully covers all feasible scenarios.

The watermark is still recoverable even after double JPEG compression with image cropping in-between. The average symbol recovery rate η for such a scenario is shown in Fig. 9d. This plot corresponds to the highest considered embedding strength without any content adaptation. The rates are averaged over different grid misalignment vectors.

3.3 Coding efficiency and effective capacity

The purpose of this experiment is to evaluate the performance of the available fountain coding variants in the considered application scenario. We focus on the

achievable overhead and the expected decoding performance for typical message lengths. In the experiment, a random message has been encoded and decoded 10,000 times, each time with a different random code. During the decoding process, we have measured the average output symbol degree and the number of output symbols which were needed for successful decoding.

The decoding performance is expressed by an abstract complexity measure θ, which corresponds to the average number of elementary decoder operations which need to be performed. Following [2] we calculate it according to:

$$\theta = \lfloor K \left(1 + \overline{\beta}\right) \overline{d} \rfloor \tag{4}$$

where K is the number of message symbols, $\overline{d}$ is the average output symbol degree and $\overline{\beta}$ is the average fountain coding overhead.

The achieved results (Table 1) confirm the expected recommendations towards the applicability of the implemented coding variants. The distribution for the LT code have been generated with the following parameters: $\eta = (0.083, 0.487, 0.032)$ for OPTD and $\delta = 0.05, c = 0.03$ for both RSD and RRSD. If the computational complexity of RLF is too high, one should use the LT code with either the original RSD or with the OPTD in case the message is short. If the computational complexity is still too high, one might consider the RRSD.

In addition to the average overhead $\overline{\beta}$ we also present the 95% overhead β_{95}, which represents the additional number of symbols for a 5% message decoding error probability. It is calculated from a cumulative histogram of the necessary output

Table 1 The overhead and expected decoding complexity for various message lengths for the LT code with different degree distributions

Dist.	Payload [B]	K	$\overline{d}$	$\overline{\beta}$ (%)	β_{95} (%)	θ
RSD	375	50	7.22	46.87	98	530
OPTD			4.93	29.76	58	320
RRSD			4.47	42.71	116	319
RSD	750	100	8.42	37.8	79	1161
OPTD			5.52	21.98	38	735
RRSD			5.14	31.62	78	676
RSD	1,875	250	10	26.10	48.8	3153
OPTD			6.33	16.14	26.8	1,838
RRSD			6.05	21.24	47.2	1,832
RSD	3,750	500	11.23	19.76	35.6	6,727
OPTD			6.96	13.27	23	3,942
RRSD			6.73	16.07	33.8	3,908
RSD	7,500	1,000	12.44	14.83	25	1,4284
OPTD			7.58	11.29	21	8,436
RRSD			7.42	12.64	26.9	8,357
RSD	15,000	2,000	13.75	11.2	18.15	30,583
OPTD			8.2	10.03	19.4	18,045
RRSD			8.13	10.32	22.45	17,943
RSD	37,500	5,000	15.36	7.67	11.58	82,688
OPTD			9.03	9.48	20.92	49,430
RRSD			9.06	8.24	20.82	49,014

symbols. 4 exemplary histograms of the necessary output symbols for two exemplary message lengths are shown in Fig. 10.

For the RLF, the expected overhead can be easily estimated from the upper bound on the decoding success probability 2^{-E} where E is the number of additional symbols. Let $\rho < 1$ denote the desired probability of decoding the message. Then, given the number of input symbols K, the number of symbols which are necessary for successful decoding is determined by (5).

$$K' = K + E = K + \log_2 \left(1 / \left(1 - \rho \right) \right) \tag{5}$$

What is important, is that the overhead for the RLF does not depend on the message length and that it asymptotically decreases to 0. At the same time, the computational complexity of the scheme increases as $O(NK^2)$ and becomes impractical for long messages.

On the other hand, the LT code which can be efficiently decoded with BP has significantly larger overhead, especially for short messages. The overhead also decreases asymptotically, yet not to 0. It is expected that in a practical scenario, for relatively long messages the average overhead can be made as small as 5% of the message length [12]. This known result is also reflected in the obtained results (Table 1).
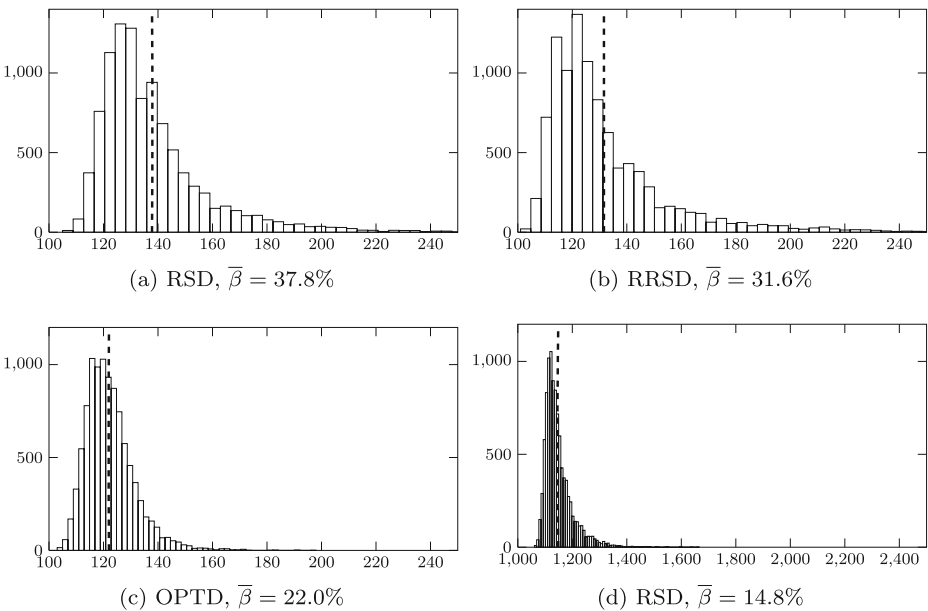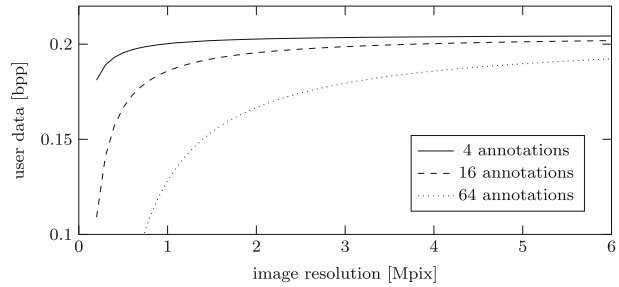

Fig. 10 Exemplary overhead histograms for successful decoding of a message: **a** RSD : $K = 100$; **b** RRSD : $K = 100$; **c** OPTD : $K = 100$; **d** RSD : $K = 1000$

Fig. 11 The estimated user data payload for message decoding failure probability 10^{-6} and several annotations in the image

The proposed scheme embeds 60 bits of watermark payload in each 256-bit macro block. Considering the necessary headers in the configuration blocks, this results in the payload of ≈ 0.22 bpp. Not all of this capacity can be used to carry user data as the utilized fountain code introduces additional overhead. The annotations are encoded independently and the overhead of each individual annotation needs to be taken into account.

Given that the desired failure probability of decoding a message is 10^{-6}, the RLF code requires additional 20 symbols per stream. Depending on the amount of embedded streams and the resolution of the image, the proposed scheme allows to embed up to 0.205 bpp of user payload. Figure 11 shows the behavior of the effective user data payload for exemplary cases of 4, 16 and 64 annotations.

4 Discussion and conclusions

In this paper we have presented a novel annotation watermarking scheme. It is based on fundamentally different assumptions than existing schemes, i.e., it has been designed with higher user payload in mind and with relaxed watermarking transparency requirements.

We have designed a layered architecture of the system and we have adopted the fountain coding paradigm for encoding the watermark payload. This approach allows to achieve straightforward multiplexing of independent data streams and overcome the problems with the non-shared selection channel which occur when dealing with cropping or when incorporating content adaptivity. Analogous approach could be used when designing systems which required support for multiple watermarks.

Our scheme is robust against cropping and lossy JPEG compression and allows to achieved significantly higher watermark payloads than existing systems. The cost of this improvement is lower fidelity of the watermarked images. Due to noise-like character of the introduced distortion, the final image quality as perceived by human eye is still very high and can be further controlled by dedicated system parameters.

One of the further research topics is related to optimization of the error correction and detection performance. The utilized 16-bit hash might not provide sufficiently low collision probability ($\approx 1.5 \cdot 10^{-5}$). A possible solution is to use more advanced

Table 2 A short comparison with existing annotation watermarking schemes

Reference	Embedding method/domain	Robustness	PSNR [dB]	Payload [bpp]	Embedded info.
Sun et al. [15]	Custom / DCT + ICA	$\eta \approx 0.95$ for JPEG 50	45	0.004	Plain ACSII
Giakoumaki et al. [6]	Quantization / Haar-DWT	JPEG > 75	42	0.026	Medical annotations, robust ID
Schott et al. [13]	DDD / DFT	JPEG > 90, cropping	47	0.012	Hierarchical annotations
Zain and Clarke [16]	LSB / Spatial	JPEG > 60	40	0.016	Medical annotations Authentication data
He et al. [7]	SS + regional statistical quantization / LBT	$\eta \approx 0.8$ for JPEG 10 $\eta \approx 1.0$ for JPEG > 50 cropping	46	32 bits = const	32-bit image ID
Our	QIM / DCT + A-SS / spatial	$\eta \approx 1.0$ for JPEG 80, cropping	38	0.22	Hierarchical annotations

fountain codes, like Raptor Codes [14] which use a carefully tailored concatenation of the LT and LDPC codes.

Table 2 summarizes existing research on annotation watermarking. The proposed scheme provides a valid alternative for applications with different requirements than commonly assumed low-payload and high-transparency.

References

1. Chen B, Wornell G (2001) Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. IEEE Trans Inf Theory 47(4):1423–1443. doi:10.1109/18.923725
2. Chen Z, Zhou Q (2010) Implementation of lt codes with a revised robust soliton distribution by using kent chaotic map. In: 2010 International Workshop on Chaos-Fractals Theories and Applications (IWCFTA), pp 149–153. doi:10.1109/IWCFTA.2010.77
3. Cox IJ (2008) Digital watermarking and steganography. Morgan Kaufmann Publishers, San Francisco
4. Eggers J, Bauml R, Tzschoppe R, Girod B (2003) Scalar costa scheme for information embedding. IEEE Trans Signal Process 51(4):1003–1019. doi:10.1109/TSP.2003.809366
5. Fridrich J (2010) Steganography in digital media—principles, algorithms and applications. Cambridge University Press, Cambridge
6. Giakoumaki A, Pavlopoulos S, Koutouris D (2003) A medical image watermarking scheme based on wavelet transform. In: Engineering in medicine and biology society, 2003. Proceedings of the 25th annual international conference of the IEEE, vol 1, pp 856–859. doi:10.1109/IEMBS.2003.1279900
7. He S, Kirovski D, Wu M (2009) High-fidelity data embedding for image annotation. IEEE Trans Image Process 18(2):429–435. doi:10.1109/TIP.2008.2008733
8. Hyytiä E, Tirronen T, Virtamo J (2006) Optimizing the degree distribution of LT codes with an importance sampling approach. In: 6th International Workshop on Rare Event Simulation RESIM 2006. Bamberg, Germany. http://www.netlab.tkk.fi/tutkimus/abi/publ/lt-resim-2006.pdf
9. Kalker T, Depovere G, Haitsma J, Maes M (1999) A video watermarking system for broadcast monitoring. In: SPIE: security and watermarking of multimedia contents, vol 3657, pp 103–112
10. Korus P, Białas J, Olech P, Dziech A (2011) A high-capacity annotation watermarking scheme. In: Multimedia communications, services and security, communications in computer and information science, vol 149. Springer, Heidelberg, pp 1–9
11. Luby M (2002) Lt codes. In: Proc 43rd ann IEEE symp on foundations of computer science, pp 16–19
12. MacKay D (2005) Fountain codes. IEE Proc Commun 152(6):1062–1068
13. Schott M, Dittmann J, Vielhauer C (2009) Annowano: an annotation watermarking framework. In: Proceedings of 6th International Symposium on Image and Signal Processing and Analysis, 2009 (ISPA 2009), pp 483–488
14. Shokrollahi A (2006) Raptor codes. IEEE Trans Inf Theory 52(6):2551–2567. doi:10.1109/TIT.2006.874390
15. Sun G, Sun H, Sun X, Bai S, Liu J (2007) Combination independent content feature with watermarking annotation for medical image retrieval. In: Second International Conference on Innovative Computing, Information and Control, 2007 (ICICIC '07), p 607. doi:10.1109/ICICIC.2007.216

16. Zain J, Clarke M (2005) Reversible watermarking surviving jpeg compression. In: 27th Annual International Conference of the Engineering in Medicine and Biology Society, 2005 (IEEE-EMBS 2005), pp 3759–3762. doi:10.1109/IEMBS.2005.1617302

Paweł Korus is a Ph.D. student at the Department of Telecommunications of AGH University of Science and Technology. His research interests include digital watermarking, multimedia security and multimedia communication systems. He has received his M.Sc. degree with honors in 2008 from the AGH University of Science and Technology. Since then, he has actively participated in both national and international research projects like BONE, INDECT and INSIGMA.

Jarosław Białas has received his M.Sc. degree from the Department of Automatics of AGH University of Science and Technology in 2011. He is currently a Ph.D. student at the Department of Telecommunications of AGH University of Science and Technology. His research interests include signal processing, biometric systems and multimedia security. He is actively participating in two research projects INDECT and INSIGMA.

Andrzej Dziech holds the position of full professor in AGH University of Science and Technology in Krakow, Poland, Department of Telecommunications. He received his M.Sc. and Ph.D. degrees from the Institute of Electrical Engineering in Saint Petersburg in 1970 and 1973, respectively. He is an author of nearly 140 publications and 6 books. His fields of interest are related to digital communication, image and data processing, data compression, information and coding theory, random signals, computer communications networks and signal processing. He was awarded 4 times by Ministry of Education of Poland for his research achievements. Professor Dziech has actively participated in the numerous international research projects, e.g. Tempus, Knixmas, Calibrate. He is currently coordinating a FP7 project INDECT.