Towards copy-evident JPEG images

Andrew B. Lewis, Markus G. Kuhn

University of Cambridge

Abstract: We present a technique for adding a high-frequency pattern to JPEG images that is imperceptible to the unaided eye, but turns into a clearly readable large-letter warning if the image is recompressed with some different quality factor. Our technique aims to achieve a goal similar to copy-evident security-printing techniques for paper documents, namely to have an easily recognizable text message appear in lower-quality copies, while leaving the visual appearance of the original unharmed. We exploit non-linearities in the JPEG process, in particular the clipping of the result of the inverse discrete cosine transform.

1   Introduction

Some security documents are printed with a carefully adjusted "copy evident" background pattern that looks uniform to the unaided eye, but will show a clearly visible large-letter warning, like "COPY" or "VOID", after having been photocopied (Figure 3). This differs from some other security-printing techniques, which rely on tools to decode the hidden message.

Screen trap and scan trap printing techniques use periodic screens of arbitrarily shaped image elements (such as dots and lines) as a carrier, into which an invisible message is modulated [vR02]. Screen traps cause interference with colour separation screens in the printing process, and scan traps cause aliasing when sampled by a digital scanner. Various types of modulation may be used: the phase, angle, size, frequency, shape or colour of
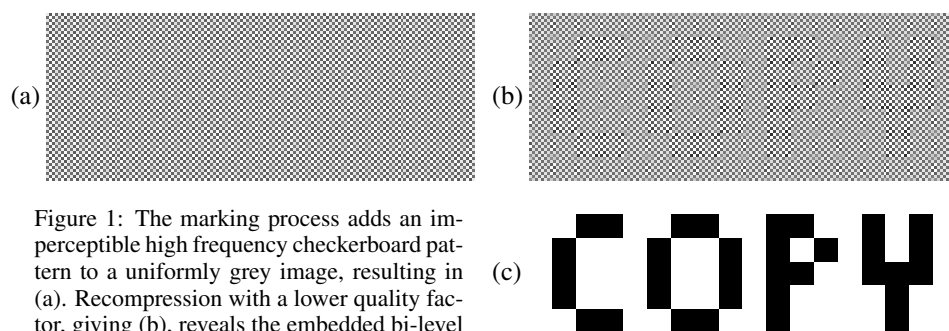

(a)

(b)

(c)

Figure 1: The marking process adds an imperceptible high frequency checkerboard pattern to a uniformly grey image, resulting in (a). Recompression with a lower quality factor, giving (b), reveals the embedded bi-level message (c).
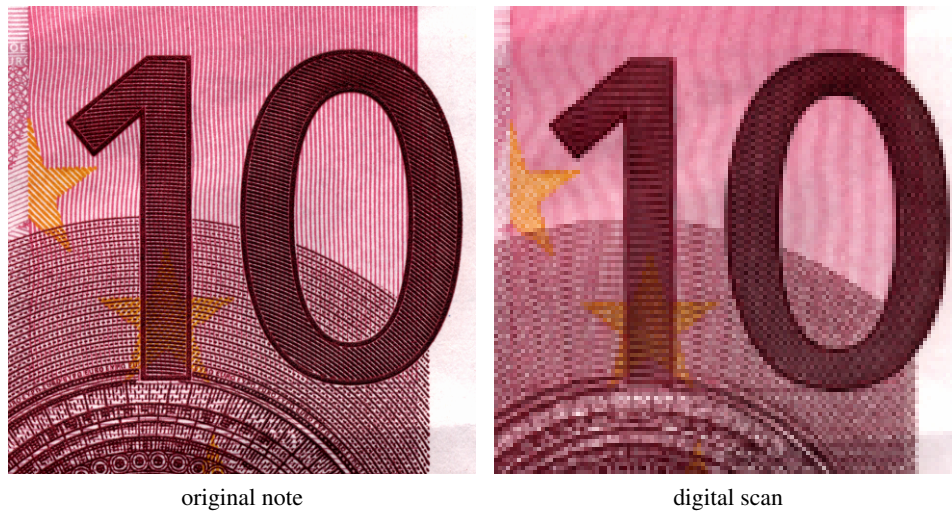
| original note | digital scan |

Figure 2: The top-right corner of the 10 EUR note is printed with a scan trap, which causes aliasing when captured at low resolutions with a digital scanner (right).

screen elements may be altered to encode the cover and covert message in the security printed image.

One such technique is screen-angle modulation (SAM) [Spa96, Spa00]. The screen elements for this technique are very fine lines, whose orientation is locally modulated with a covert image. The chosen image becomes visible after copying. It is also possible to hide a covert image within a cover image, rather than in a uniform region, by modulating the width of the screen elements with the cover image.

Further information on security-printing techniques is available in [vR02, vR05]. In general, they cause nearly invisible high-frequency differences in the image signal to turn into clearly visible low-frequency components.

Are similar techniques possible with digital formats? Can we add some suitably crafted security patterns to digital images, videos, or audio recordings that remain imperceptible
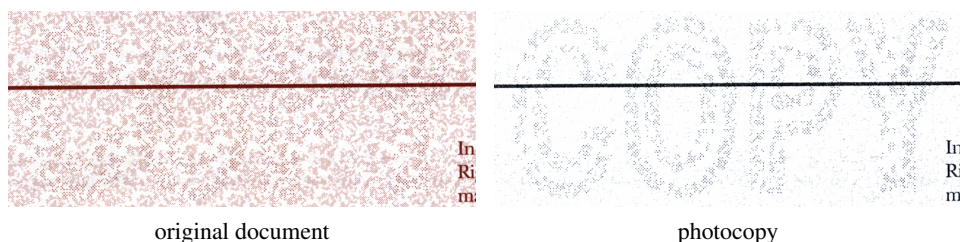

| original document | photocopy |

Figure 3: The background pattern of this academic transcript was printed using two different dot sizes, adjusted to result in the same halftone (left). The photocopier fails to reproduce the smaller dots, leaving the regions printed with larger dots to form a visible "COPY" warning (right).

in the original output of the marking process, but are likely to become visible (or audible) when standard lossy processing techniques are applied, such as requantization or resampling? Can we create test images that output a human-readable warning message if the image quality may have been degraded by some hidden processing step, for example, if an Internet service provider recompresses JPEG images on web pages to a lower quality factor to make the HTTP connection appear faster?

This is not an easy problem. Commonly used encoding and processing algorithms have been designed specifically to minimize perceptible distortion. Therefore, we need to understand exactly what distortions are left and how to maximize them. How can we make the distortion invisible in the original marked version, yet visible in processed derivative copies? One approach is to add signals to the marked copy that are carefully balanced to cancel out each other's distortions, hoping that any further processing will destroy that balance. This might involve generating a compressed representation that, even though it decompresses correctly, could never have been generated by a normal compression step. It might also involve exploiting non-linearities (quantization, clipping, gamma correction) or artifacts (aliasing, blocking) that compression algorithms sometimes introduce.

Ideally, we would even like to have control over the conditions under which the embedded mark becomes visible. In some applications we prefer a *targeted mark*, which only becomes visible when one particular a priori known processing step is applied. (Imagine a video that shows a warning if it has been uploaded to a particular website where all material is recompressed with fixed settings.) In other applications, we might prefer an *untargeted mark*, which becomes visible with high probability as soon as any of a number of possible processing parameters are applied.

This paper describes the result of our initial exploration of this area, namely the generation of a JPEG image of uniform colour, which on recompression with a particular quantization table will result in a message appearing.

2   JPEG

To produce suitable marks for JPEG images, we need to take into account the processing steps which produce the JPEG bitstream.

The compressor takes an image with pixels in the RGB colour space, transforms these input samples to a colour space with Y (luma), Cb and Cr (chroma) components, optionally downsamples the chroma channels by a factor of two in one or both directions and calculates the discrete cosine transform (DCT) of all 8×8 non-overlapping blocks in each channel independently, to give a representation for each block as a sum of scaled, sampled cosines. For each spatial frequency in a block, the associated coefficient $X_{i,j}$ is then linearly quantized by a factor provided in a quantization table $Q \in \mathbb{N}^{8 \times 8}$, which determines with what precision the amplitude of that frequency component is represented. Information is intentionally discarded during chroma subsampling (if enabled) and quantization of DCT coefficients. Subsequent steps code the latter without further loss of information.

Decompression inverts each step in turn, performing dequantization, inverse DCT, chroma

upsampling (if required) and conversion back to the RGB colour space.

The quantization factors in Q are given as an input parameter to the compressor, calculated from a scalar *quality factor* (e.g., $1, \ldots, 100$) in one popular implementation [Lan]. The table Q is encoded in the header of the JPEG file so that the decompressor has it available for dequantization.

A block's frequency-domain coefficient $X_{i,j}$ ($i, j \in \{0, \ldots, 7\}$) corresponds to the amplitude of the product of two cosine waves, one with $i/2$ cycles per block vertically and one with $j/2$ cycles per block horizontally. It is quantized by a factor $Q_{i,j}$ to give

$$\hat{X}_{i,j} = \operatorname{sgn}\left(X_{i,j}\right) \cdot \left\lfloor \frac{|X_{i,j}| + \lfloor Q_{i,j}/2 \rfloor}{Q_{i,j}} \right\rfloor. \tag{1}$$

The corresponding dequantization operation in the decompressor multiplies the quantized coefficient by the quantization factor (from the table in the JPEG header):

$$X'_{i,j} = Q_{i,j} \cdot \hat{X}_{i,j} \tag{2}$$

Blocks with $\hat{X}_{i,j} = 0$ for high values of i and j (the perceptually less important higher frequencies) are coded very compactly in the final lossless stage. Therefore, in practice, most quantization tables have high values $Q_{i,j}$ for high values of i and j. For example, the Independent JPEG Group (IJG [Lan]) codec implementation's default quantization table ("quality factor" 75, the same table as that recommended by the JPEG standard for providing a 'nearly indistinguishable' reconstruction of the original [ISO, Annex K.1]) is

$$Q = \begin{pmatrix} 8 & 6 & 5 & 8 & 12 & 20 & 26 & 31 \\ 6 & 6 & 7 & 10 & 13 & 29 & 30 & 28 \\ 7 & 7 & 8 & 12 & 20 & 29 & 35 & 28 \\ 7 & 9 & 11 & 15 & 26 & 44 & 40 & 31 \\ 9 & 11 & 19 & 28 & 34 & 55 & 52 & 39 \\ 12 & 18 & 28 & 32 & 41 & 52 & 57 & 46 \\ 25 & 32 & 39 & 44 & 52 & 61 & 60 & 51 \\ 36 & 46 & 48 & 49 & 56 & 50 & 52 & 50 \end{pmatrix}. \tag{3}$$

3   Marking method

We now describe our method for creating a JPEG file with an embedded targeted mark, one that will become visible after recompression with a *known* quantization table Q'. We embed a single pixel (one bit, foreground or background) of the marking message in each 8×8 luma DCT block (Figure 1). We replace each such block with an equivalent looking block that contains an added high-frequency checkerboard dither pattern. We choose the amplitude of that dither pattern such that half its pixel values end up close to the clipping limit (0 or 255). The exact amplitude chosen differs depending on whether the block represents a foreground or background pixel of the marking message. We choose this pair of amplitudes such that their values are (a) as close together as possible, (b) rounded in

opposite directions after requantization with Q', and (c) such that half of the pixels in a requantized foreground block will *exceed* the clipping limit after the inverse DCT in the decoder (Figure 4). As a result, the clipping operation in the decoder will affect the average pixel value in foreground blocks, but not in background blocks, leading to a visible difference.
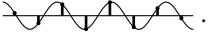
In a JPEG encoder, each quantization decision boundary

$$B_{i,j}(n) = \frac{1}{2}(n \cdot Q_{i,j} + (n - \mathrm{sgn}(n)) \cdot Q_{i,j}) = Q_{i,j} \cdot (n - \mathrm{sgn}(n)/2) \qquad (4)$$

lies centred between two adjacent multiples of quantization factor $Q_{i,j}$, where $n \in \mathbb{Z} \backslash \{0\}$. The pair of consecutive integers $|X_{i,j}^{\top}(n) - X_{i,j}^{\perp}(n)| = 1$ on either side of this boundary map to adjacent integers n and $n - \mathrm{sgn}(n)$, respectively, when quantized:

$$
\begin{aligned}
X_{i,j}^{\top}(n) &= n \cdot Q_{i,j} - \mathrm{sgn}(n) \cdot \lfloor Q_{i,j}/2 \rfloor \\
&= B_{i,j}(n) + (Q_{i,j} \bmod 2) \cdot \mathrm{sgn}(n)/2, \\
X_{i,j}^{\perp}(n) &= n \cdot Q_{i,j} - \mathrm{sgn}(n) \cdot (\lfloor Q_{i,j}/2 \rfloor + 1) \\
&= X_{i,j}^{\top}(n) - \mathrm{sgn}(n).
\end{aligned} \qquad (5)
$$

A DCT coefficient taking on one of these values will incur the maximum quantization error when compressed. For a particular DCT coefficient position (i, j), if we compress two blocks, one using $X_{i,j}^{\top}(n)$ and the other using $X_{i,j}^{\perp}(n)$, these will each experience maximum quantization error, but in opposite directions, despite the fact that the uncompressed appearance of the two blocks is very similar. Figure 4 shows this effect where the first compression uses a low quantization factor $Q_{i,j} = q_0$ and the second uses a high quantization factor $Q'_{i,j} = q_1$ (harsher quantization).

To embed a binary message (such as the text "COPY") in the cover image, we map each pixel in the message to an 8×8 block in the cover, and set the amplitude of a particular DCT coefficient position (i, j) to $X_{i,j}^{\top}(n)$ in foreground blocks and $X_{i,j}^{\perp}(n)$ in background blocks when quantized in the marked original with q_0. To make this effect as noticeable as possible, we choose the coefficient (i, j) so that the associated recompression quantization factor $q_1 = Q'_{i,j}$ is large. $X_{7,7}$ is the highest spatial frequency component and normally uses a large quantization factor. This coefficient's frequency component corresponds in the spatial domain to a windowed checkerboard pattern ▓; the associated 1-D sampled cosine basis vector is ⌁⌁⌁⌁⌁ .

A 2-D checkerboard pattern will be perceived with a brightness approximately equal to its mean value (subject to gamma correction), and two checkerboard patterns with the same mean but different amplitudes will be almost indistinguishable.

However, we wish to introduce contrast between blocks in a more perceptually important low frequency. The results of the inverse DCT are clipped so that they lie in the range $\{0, \ldots, 255\}$. If we arrange, by suitable choice of n, for some of the spatial domain image samples in foreground message blocks to exceed 255 after recompression with Q', these values will be clipped, while the lower values in the checkerboard pattern will not be clipped. Similarly, sample values less than 0 will be clipped after recompression. The
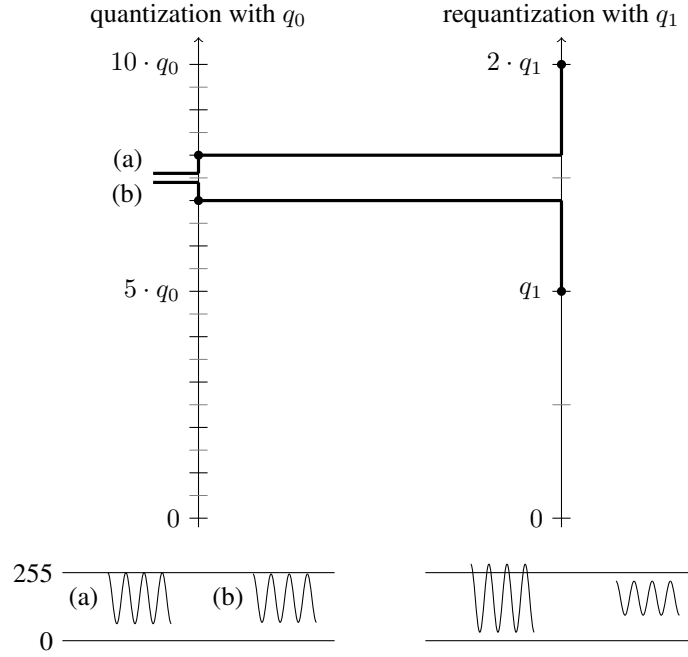
Figure 4: The quantization of two values (a) and (b) for a 3.5 cycles/block frequency component, first with quantization factor q_0, then at a lower quality with quantization factor q_1. The results of the inverse transform when the block is combined with a DC component equivalent to 192 are shown one dimensionally. Note that the higher amplitude signal (a) will be clipped after recompression, which reduces its mean.

perceived brightness of the foreground block will, therefore, be reduced (or increased) compared to a block corresponding to a background pixel in the message, where no clipping will occur: the balance of high and low samples in the checkerboard pattern will be destroyed in the recompression step. Figure 5 demonstrates this effect.

This results in a low-frequency contrast between foreground and background blocks, leading to a visible message in the recompressed version. In the marked original, we set $q_0 = Q_{7,7}$ as small as possible while still providing a slight difference in the amplitude of the checkerboard pattern between foreground blocks and background blocks in the spatial domain, and make sure that the amplitudes are on either side of a quantization boundary (using the amplitudes $X_{i,j}^\top(n)$ and $X_{i,j}^\perp(n)$, from (5)). Writing the bitstream directly, rather than using a JPEG compressor, allows for exact control over coefficient values and the quantization table required to do this.

Some combinations of block values and target quantization matrices lead to unmarkable blocks, for example, if addition of a checkerboard pattern of amplitude $X_{7,7}^\top(n)$ to the original block causes it to clip already (i.e. the value for $X_{7,7}$ which would just cause clipping lies between a multiple of the requantization factor and the next higher quantization decision boundary), then this will cause unbalanced distortion in the marked original.
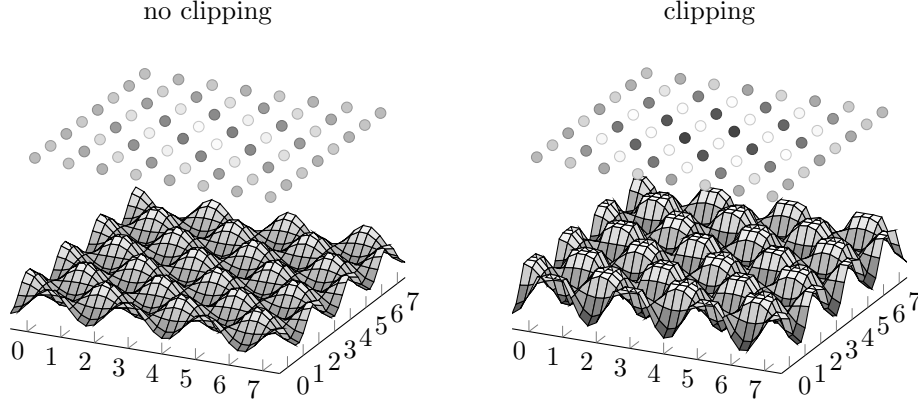
1587

Figure 5: $\min(255, a \cdot \cos(\pi k) \cdot \cos(\pi n) + b)$ plotted with (a) $a = 64$ and $b = 192$, where no clipping occurs, and (b) $a = 128$ and $b = 192$, where half the outputs are clipped to 255. Block (a) has a higher mean value than block (b), and therefore appears brighter.

Because the $X_{7,7}$ component corresponds to a windowed checkerboard pattern (sampling introduces a low beat frequency), the block will not appear as a uniform checkerboard pattern after recompression.

Our marking process is shown in Algorithm 1. Given an 8×8 block of DCT coefficients from the original image B, the binary value of the message m and the target quantization table Q', MARKBLOCK(B, m, Q') searches through the possible amplitudes x for the checkerboard pattern and returns either FAIL (for unmarkable blocks), or a replacement image block with an added checkerboard pattern at the amplitude necessary to cause clipping after recompression with Q'. One value for the pattern's amplitude is tested on each iteration, with the current higher amplitude candidate marked block stored in $H[x]$ (returned when $m = 1$), and the previous iteration's marked block stored in $H[x - 1]$ (returned when $m = 0$). The function returns FAIL for blocks which cannot be marked if (1) the addition of the checkerboard pattern causes clipping, (2) clipping occurs in the spatial domain block $h = \text{IDCT}(H)$ before recompression, or (3) clipping occurs only after recompression but the highest frequency coefficient (representing a checkerboard pattern) has not changed.

To mark a natural image, rather than a uniform region, we must replace blocks with perceptually similar checkerboard patterns of the same brightness. However, *pixel values* from $\{0, \ldots, 255\}$ are not proportional to actual display *brightness* (photons per second), but instead are related by a power law (gamma correction): a pixel value of s results in a pixel brightness proportional to s^γ, where the constant γ is the exponent for the display device (typically $\gamma \approx 2.2$).

To find the checkerboard pattern's mean pixel value μ for a given amplitude x (in the image sample domain) such that its brightness matches that of the original block m^γ, we

---
Algorithm 1 Marking algorithm for JPEG image blocks

$\text{DCT}(b)$ returns the discrete cosine transform of block b.
$\text{IDCT}(B)$ returns the inverse discrete cosine transform of block B.
$\text{CLIPS}(b)$ returns true if any sample in b exceeds 255 or is less than 0.
$\text{QUANTIZE}(B, Q)$ quantizes B using table Q according to Equation (1).
$\text{DEQUANTIZE}(B, Q)$ dequantizes B using table Q according to Equation (2).
$\text{CHECKERBOARD}(x)$ returns an 8×8 checkerboard pattern with elements $+x$ and $-x$.

$H[x]$ stores the candidate DCT coefficient block, with spatial domain representation $h[x]$. $\hat{H}[x]$ and $\hat{h}[x]$ are those same blocks after requantization with Q'.

1: **function** $\text{MARKBLOCK}(B \in \mathbb{Z}^{8 \times 8}, m \in \{0, 1\}, Q' \in \mathbb{N}^{8 \times 8})$
2:     **for** $x \leftarrow 1$ to 128 **do** $\triangleright$ For each amplitude value x
3:          $h[x] \leftarrow \text{IDCT}(B) + \text{CHECKERBOARD}(x)$
4:          **if** $\text{CLIPS}(h[x])$ **then**
5:              **return** FAIL_1 $\triangleright$ The checkerboard signal is out of range
6:          **end if**
7:          $H[x] \leftarrow \text{DCT}(h[x])$
8:          **if** $\text{CLIPS}(\text{IDCT}(H[x]))$ **then**
9:              **return** FAIL_2 $\triangleright$ The original marked block must not clip
10:         **end if**
11:         $\hat{H}[x] \leftarrow \text{DEQUANTIZE}(\text{QUANTIZE}(H[x], Q'), Q')$
12:         $\hat{h}[x] \leftarrow \text{IDCT}(\hat{H}[x])$
13:         **if** $\text{CLIPS}(\hat{h}[x])$ and $x > 1$ **then**
14:             **if** $\hat{H}[x]_{7,7} \neq \hat{H}[x-1]_{7,7}$ **then**
15:                **if** $m = 1$ **then return** $H[x]$ **else return** $H[x-1]$
16:             **else**
17:                **return** FAIL_3 $\triangleright$ Clipping occurs on recompression, but $H[x]$ and
18:             **end if** $H[x-1]$ are not either side of the quantization
19:         **end if**   boundary of the highest frequency coefficient: $\nexists n$:
20:      **end for** $X_{7,7}^{\top}(n) = H[x]_{7,7}$
21: **end function**

---

solve Equation (8) to find the brightness amplitude δ given x and m, then substitute this back into Equation (7) to find μ [Kuh03, pp. 57–60]:

$$\mu \pm x = (m^{\gamma} \pm \delta)^{\frac{1}{\gamma}} \tag{6}$$

$$\mu = \frac{1}{2}\left((m^{\gamma} + \delta)^{\frac{1}{\gamma}} + (m^{\gamma} - \delta)^{\frac{1}{\gamma}}\right) \tag{7}$$

$$x = \frac{1}{2}\left((m^{\gamma} + \delta)^{\frac{1}{\gamma}} - (m^{\gamma} - \delta)^{\frac{1}{\gamma}}\right) \tag{8}$$

If this is implemented in a function $\text{GAMMACORRECT}(m, x)$, which returns μ, it can be used to alter the checkerboard pattern added on line 3 of Algorithm 1, making the replacement blocks perceptually similar to the original blocks.

To test the marking on all possible uniform blocks, we marked a 512×512 pixel test image consisting of a grid of 64×64 non-overlapping 8×8 pixel blocks with a black to white gradient in raster-scan order, containing two horizontally adjacent blocks at each DCT-domain brightness value, to allow comparison of the cases $m = 0$ and $m = 1$ (Figure 6 (a)): the pixel at (x, y) is within a block $B^{(u,v)}$, $(u, v) = (\lfloor x/8 \rfloor, \lfloor y/8 \rfloor)$, which has one non-zero DCT coefficient taking the value $B_{0,0}^{(u,v)} = \lfloor u/2 \rfloor + 32 \cdot v - 1024$. Figure 6 shows the results of applying $\text{MARKBLOCK}(B^{(u,v)}, u \bmod 2, Q')$, where Q' is the quantization table for IJG quality factor 50, to this test image (b) before and (c) after recompression, where unmarkable blocks have been replaced with a digit indicating the type of failure.

4   Conclusion

We presented a first demonstration of a copy-evident multi-media file, in which a human-readable message becomes visible after recompressing the original. The technique now needs to be extended to handle arbitrary photographs, not just uniform regions. More work is also needed to develop an untargeted JPEG marker that is not tied to a particular quantization table Q', but results in a readable message with high probability over a range of quality factors.

References

[ISO]   ISO/IEC 10918-1:1994, Information technology – Digital compression and coding of continuous-tone still images: Requirements and guidelines. International Organization for Standardization.

[Kuh03] Markus G. Kuhn. Compromising emanations: eavesdropping risks of computer displays. Technical Report UCAM-CL-TR-577, University of Cambridge, Computer Laboratory, December 2003.

[Lan]   Thomas G. Lane. Independent JPEG Group library. *http://www.ijg.org*.

[Spa96] Sijbrand Spannenburg. Optically and machine-detectable copying security elements. In *Proceedings of SPIE*, volume 2659, page 76, 1996.

[Spa00] Sijbrand Spannenburg. Developments in digital document security. In *Proceedings of SPIE*, volume 3973, page 88, 2000.

[vR02]   Rudolf L. van Renesse. Hidden and scrambled images – a review. In *Proceedings of SPIE*, volume 4677, page 333, 2002.

[vR05]   Rudolf L. van Renesse, editor. *Optical Document Security*. Artech House, 3rd ed. edition, 2005.

(a) original   (b) marked
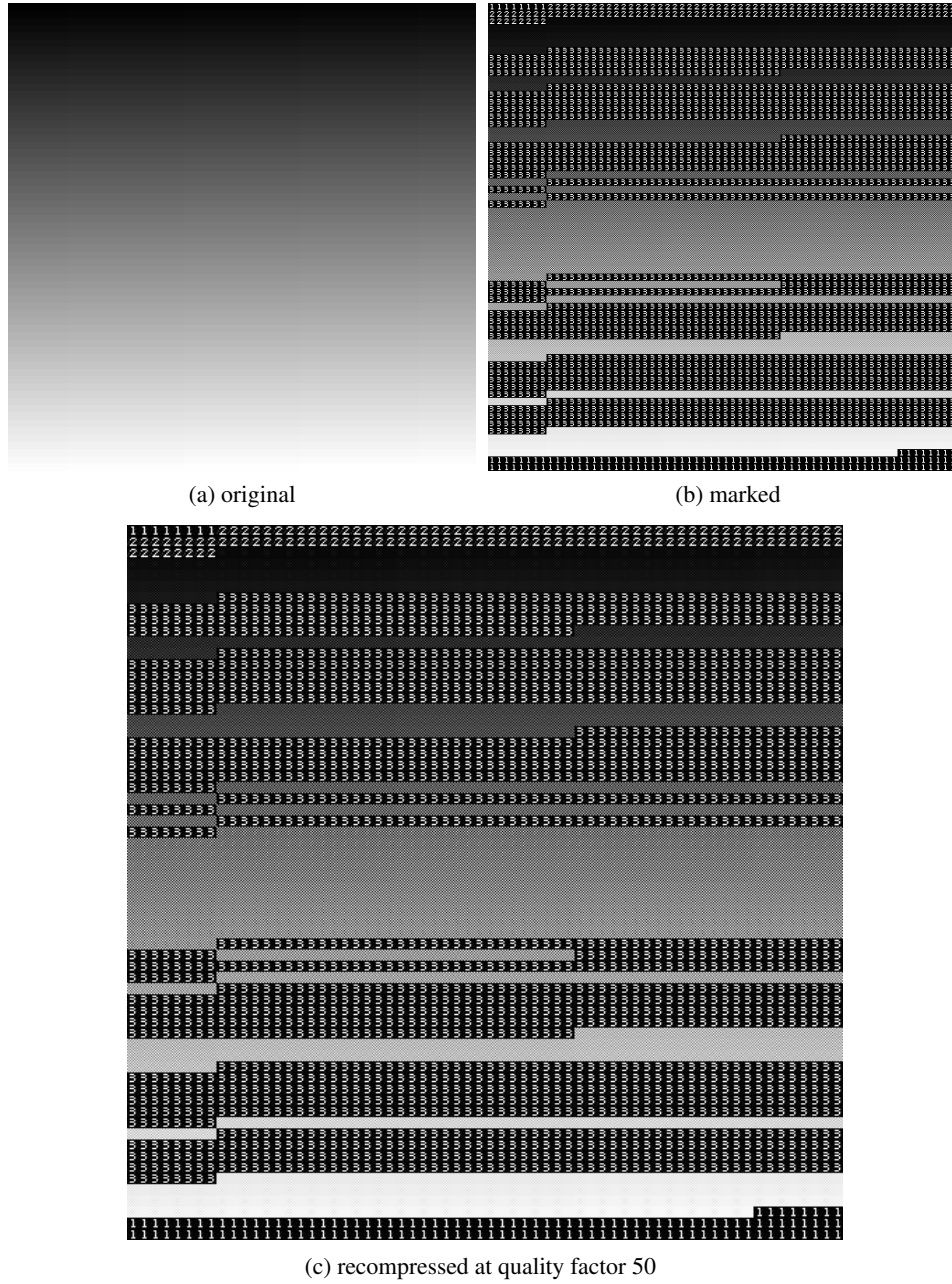

(c) recompressed at quality factor 50

Figure 6: Marking and recompression of raster-scan order black to white gradient (a) testing each DCT domain brightness value, with a repeating tiled message of (0, 1). The marked image (b) shows each block replaced by a checkerboard pattern; blocks which cannot be marked successfully are replaced with a digit corresponding to the type of error in Algorithm 1. (c) shows the result of recompression with the target quantization matrix.

1591