

Article

Reduction of Artefacts in JPEG-XR Compressed Images

Kai-Lung Hua ^{1,2}, Ho Thi Trang ¹, Kathiravan Srinivasan ^{3,*}, Yung-Yao Chen ⁴,
Chun-Hao Chen ¹, Vishal Sharma ⁵ and Albert Y. Zomaya ⁶

¹ Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan; hua@mail.ntust.edu.tw (K.-L.H.); d10515804@mail.ntust.edu.tw (H.T.T.); m10015902@mail.ntust.edu.tw (C.-H.C.)

² Center for Cyber-physical System Innovation, National Taiwan University of Science and Technology, Taipei 10607, Taiwan

³ School of Information Technology and Engineering, Vellore Institute of Technology, Vellore 632 014, India

⁴ Graduate Institute of Automation Technology, National Taipei University of Technology, Taipei 10607, Taiwan; yungyaochen@mail.ntut.edu.tw

⁵ Department of Information Security Engineering, Soonchunhyang University, Asan-si 31538, Korea; vishal_sharma2012@hotmail.com

⁶ School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia; albert.zomaya@sydney.edu.au

* Correspondence: kathiravan.srinivasan@vit.ac.in

Received: 13 February 2019; Accepted: 4 March 2019; Published: 9 March 2019



Abstract: The JPEG-XR encoding process utilizes two types of transform operations: Photo Overlap Transform (POT) and Photo Core Transform (PCT). Using the Device Porting Kit (DPK) provided by Microsoft, we performed encoding and decoding processes on JPEG XR images. It was discovered that when the quantization parameter is >1 -lossy compression conditions, the resulting image displays chequerboard block artefacts, border artefacts and corner artefacts. These artefacts are due to the nonlinearity of transforms used by JPEG-XR. Typically, it is not so visible; however, it can cause problems while copying and scanning applications, as it shows nonlinear transforms when the source and the target of the image have different configurations. Hence, it is important for document image processing pipelines to take such artefacts into account. Additionally, these artefacts are most problematic for high-quality settings and appear more visible at high compression ratios. In this paper, we analyse the cause of the above artefacts. It was found that the main problem lies in the step of POT and quantization. To solve this problem, the use of a “uniform matrix” is proposed. After POT (encoding) and before inverse POT (decoding), an extra step is added to multiply this uniform matrix. Results suggest that it is an easy and effective way to decrease chequerboard, border and corner artefacts, thereby improving the image quality of lossy encoding JPEG XR than the original DPK program with no increased calculation complexity or file size.

Keywords: JPEG XR; HD photo; Windows Media Photo; chequerboard artefacts; border artefacts; corner block artefacts

1. Introduction

In a smart city environment, a colossal quantity of image data is generated from traffic control systems, intruder detection and surveillance systems and from other intelligent sensing gadgets and devices. Therefore, sending the data across the network might consume very high network traffic; hence, these images are compressed using suitable compression approaches such as Joint Photographic Experts Group (JPEG), JPEG Extended Range (JPEG-XR) and so on [1–3]. JPEG-XR was initially presented under the name Windows Media Photo, and subsequently, it was retitled as High Definition

(HD) Photo. Further, in 2006, Microsoft proposed it as a compression method and file format for images. After that, in 2009, it was announced as the International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) and International Telecommunication Union (ITU) worldwide standard by the Joint Photographic Experts Group (JPEG) [4–6]. JPEG-XR has better compression efficiency than JPEG [7,8]. Its advantages include lossless compression, transparency, and support for multiple image formats. Moreover, at low bit rates, JPEG images usually suffer from blocking artefacts [9–11]. In comparison with JPEG 2000, JPEG-XR is less complicated; however, the bit rate for a given Peak Signal-to-Noise Ratio (PSNR) is comparable to the bit rate of JPEG-2000 and could be as much as 50 percent lower than the bit rate of JPEG [12]. Furthermore, the compression algorithm of JPEG-XR is conceptually identical to that of JPEG [13]. In the JPEG-XR process, an image is segregated into macroblocks of fixed size. These macroblocks in the spatial domain are then transformed into the frequency domain; the frequency components are then quantized and the entropy encoded [14].

The reversible integer transform is amongst the most significant features of the JPEG-XR compression standard. Further, this feature relies on two transform operators, namely the Photo Core Transform (PCT) [15] and the Photo Overlap Transform (POT). Moreover, the process involved in PCT is identical to the Discrete Cosine Transform (DCT) methodology, where the image is transformed into corresponding frequency groups [16,17]. The output image from the JPEG-XR process has blocking artefacts, which is similar to the discrete cosine transform approach. In order to overcome this limitation, the photo overlap transform is devised in such a way that it utilizes the correlation between the boundaries of the blocks for reducing or removing the blocking artefacts [18]. Moreover, the operators are devised by using estimates of rotation operations using linear shears. As a result, the chequerboard artefacts are produced due to these estimates. Furthermore, given an uncompressed image with constant-intensity areas, the output of the JPEG-XR process might result in intensity variations similar to chequerboard artefacts. Nevertheless, these artefacts cannot be prominently observed through human vision. However, it might be a severe challenge in image document processing pipelines. Assume, for instance, an auto-cropping technique can be used by a scanner for removing the regions beyond a predefined threshold value [19,20]. Auto-cropping might not happen when there is the presence of the chequerboard artefact in the output image since the intensity values are below the predefined threshold. Furthermore, in the printing process of an image, the nonlinearities are another reason for augmenting the chequerboard artefacts. Similar to these discussions, some of the applications of the inverse problems, such as joint reconstruction of kinetic parametric images [21], reconstruction of volumetric myocardial [22] and noninvasive computational imaging [23], can be followed to further understand the advantages of inverse solutions.

Furthermore, in JPEG-XR, POT and PCT are implemented by the lifting structure [24,25]. This allows transform operations to be broken down into simple integer additions and subtractions (shears). This effectively lowers the computational complexity and resources consumed during the encoding and decoding process, which is especially advantageous on mobile or embedded devices where power is often limited. This implementation also offers reversibility so that lossless encoding can be performed. Inverse transforms can be easily implemented by sign reversal and the reciprocal.

Microsoft began offering the download of the DPK (Device Porting Kit) [26] at the end of 2006 for research and development purposes. While encoding images using this tool, we discovered that when the quantization parameter is >1 -lossy compression conditions, the resulting image showed chequerboard block artefacts, border artefacts and corner artefacts. The artefacts are less obvious when the compression ratio is low (Quantization Parameter (QP) < 41). However, at a medium to high compression ratio (QP ≥ 41), they become more apparent, especially on pure colour (monochrome) images where the artefacts can be observed directly by the naked eye. While the artefacts do not significantly lower the image quality, they could cause annoyance when processing images in special situations. These could include post-processing of scans or reproduced photos, identification of characteristics or defects in computer vision, medical analysis and geographic analysis.

In Section 2 of this paper, we will analyse chequerboard block artefacts, border artefacts and corner artefacts respectively. Through implementation and observation, we will speculate about possible causes and prove them mathematically. In Section 3, we will design a separate “uniform matrix” for each of the three artefacts in order to improve encoding quality. In Section 4, we will implement the aforementioned uniform matrices and compare the results with Microsoft’s DPK.

It will become apparent that our method effectively removes chequerboard, border and corner artefacts and provides better image quality and compression ratio.

2. Chequerboard, Border and Corner Block Artefacts

We used the DPK provided by Microsoft to encode and decode images. We discovered that chequerboard, border and corner artefacts occur when the $QP > 1$. Figure 1a depicts the original image before transformation. It is a 64×64 monochrome bitmap (BMP). Figure 1b portrays a lossless transform using $QP = 1$. It is identical to the original image. Figure 1c shows a lossy transform using $QP = 41$. The difference between it and the original is not very apparent. Figure 1d illustrates a lossy transform using $QP = 61$. The chequerboard, border and corner artefacts are more obvious than in Figure 1c. Figure 2a,b shows the results of linear contrast enhancement ($[1] f_{ENC}(p[i]) = \min(p) + (p[i] - \min(p)) * R$, where p = pixel array of image, i = array-index in p , $R = 6$ (expand ratio, fixed in this paper)) in Figure 1c,d. Chequerboard, border and corner artefacts are very noticeable. Note that if we set the overlap parameter $L = 0$, the artefacts disappear even without transformation with POT, as shown in Figure 2c,d. We can infer from the above discussion that the two causes of the artefacts are:

1. lossy quantization ($QP > 1$)
2. the use of POT ($L > 0$)

Hereinafter, we will analyse each of the two causes.

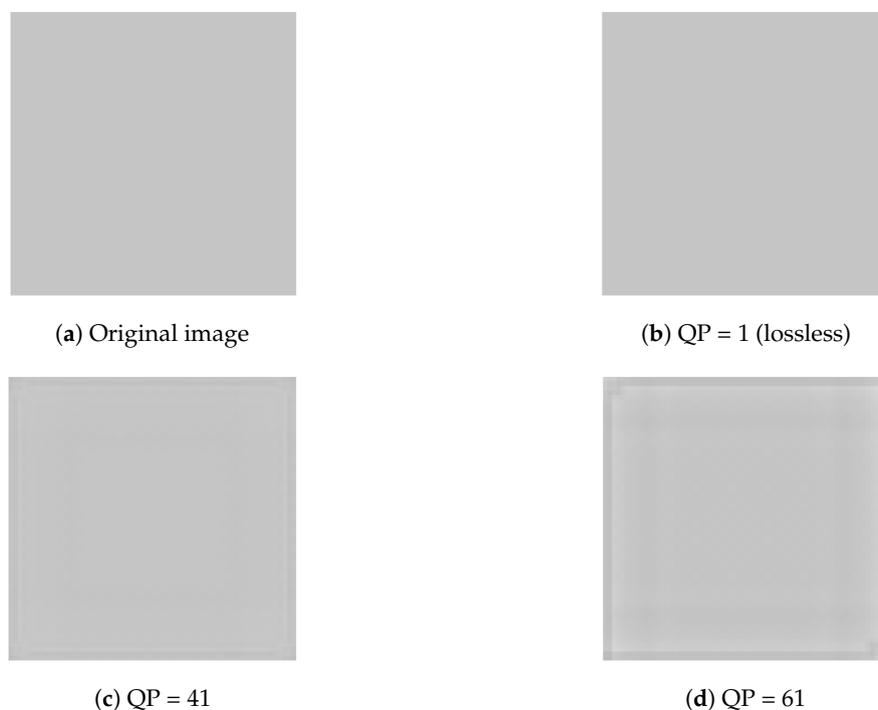


Figure 1. Part 1. Chequerboard block artefacts. QP, Quantization Parameter.

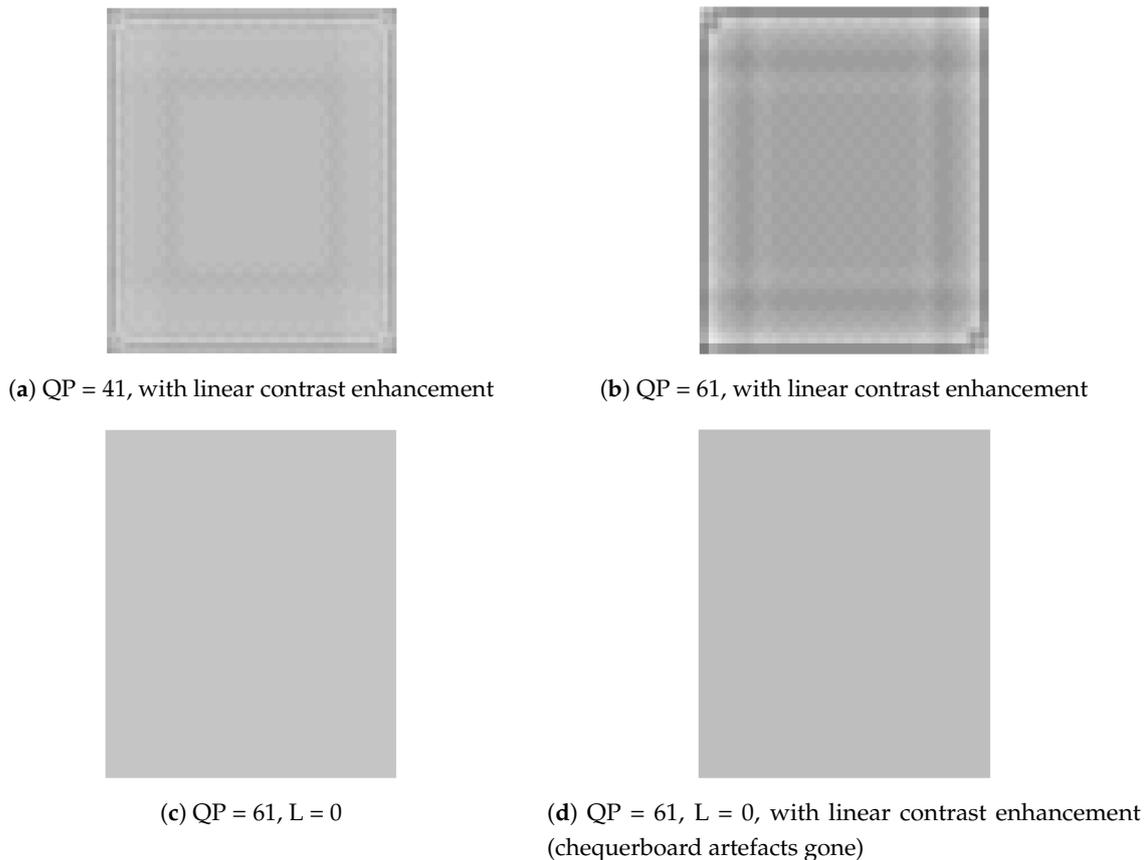


Figure 2. Part 2. Chequerboard block.

2.1. Lossy Quantization and Irreversibility

Quantization of DPK is lossless when the quantization parameter $QP = 1$, as shown in Figure 3a. When $QP > 1$, the result is similar to uniform quantization, as shown in Figure 3b–d. The solid lines are the quantization results of $QP = 21$, $QP = 41$, and $QP = 61$, respectively. It can be observed that as QP increases, the interval (step) becomes wider.

The process of JPEG-XR encoding is as shown in Figure 4. From [18], we know that the POT and PCT implemented by the lifting structure are “reversible”. This means that after a value has been transformed by POT or PCT, we can apply the corresponding inverse transforms to return it to its original value, as shown in Figure 5.

When the quantization parameter $QP = 1$, the quantization is both lossless and reversible. With $QP = 1$ and the reversibility of POT and PCT, the encoding/decoding process of JPEG-XR becomes completely lossless. This way, no chequerboard, or border, artefacts will be produced, as shown in Figure 6.

On the other hand, when $QP > 1$, or when quantization is lossy, the values may be offset or distorted. The inverse PCT, POT step is then irreversible and may cause chequerboard, border and corner artefacts, as shown in Figure 7.

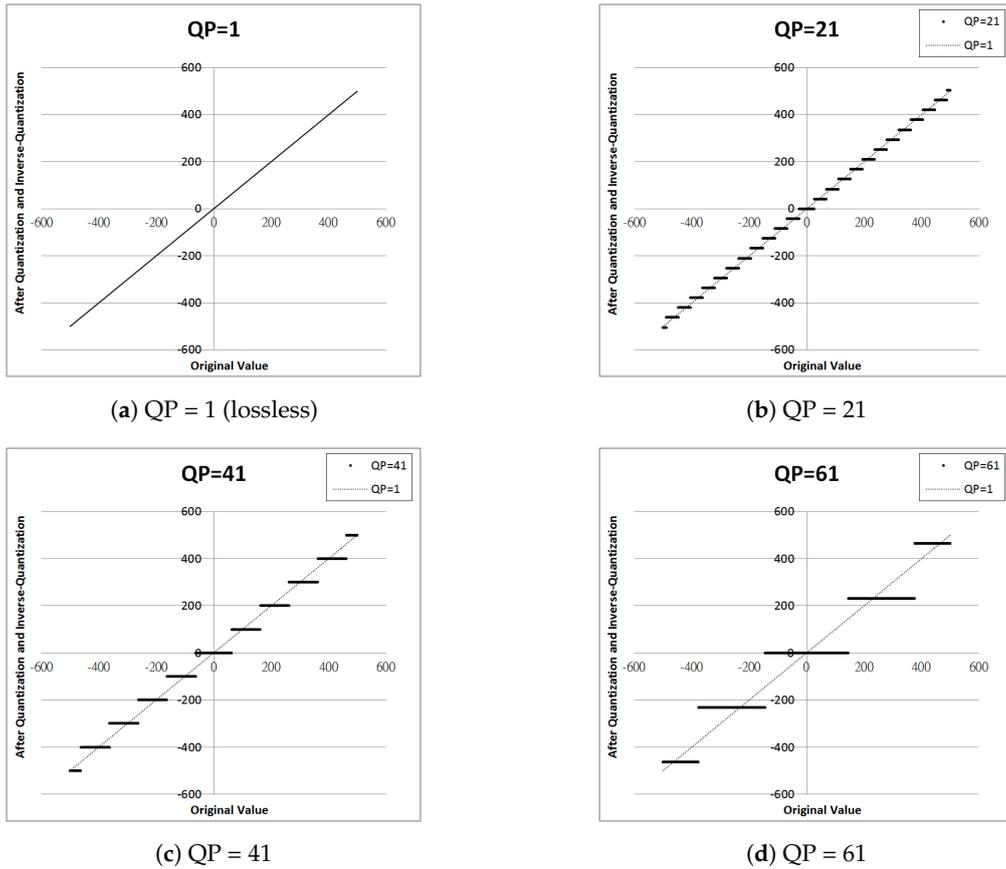


Figure 3. Lossless (QP = 1) and lossy (QP > 1) quantization.

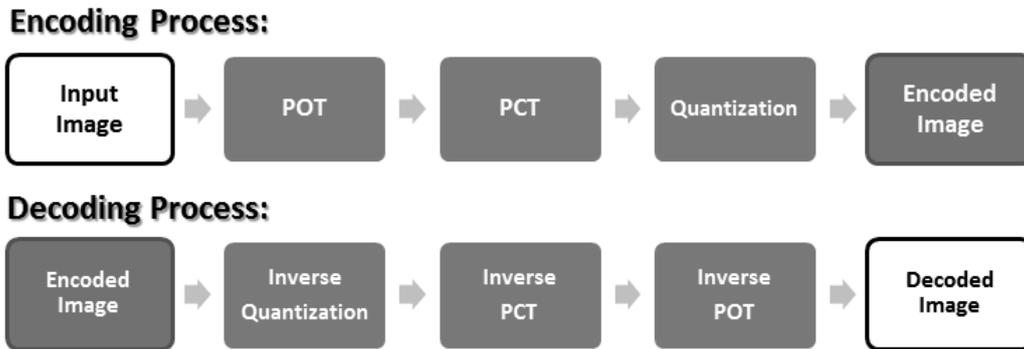


Figure 4. JPEG-XR encoding/decoding process. POT, Photo Overlap Transform; PCT, Photo Core Transform.

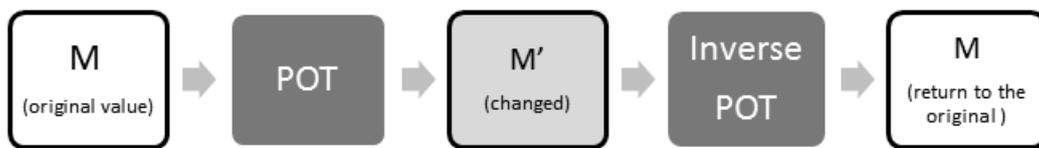


Figure 5. The “reversibility” of POT (and PCT).

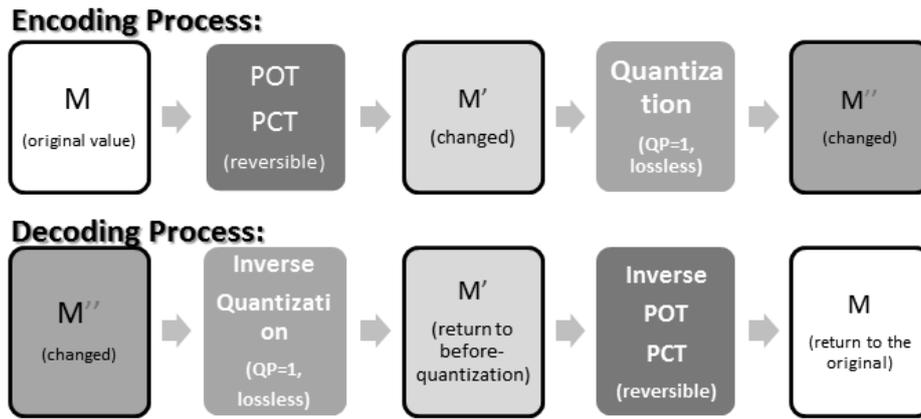


Figure 6. The “reversibility” when QP = 1.

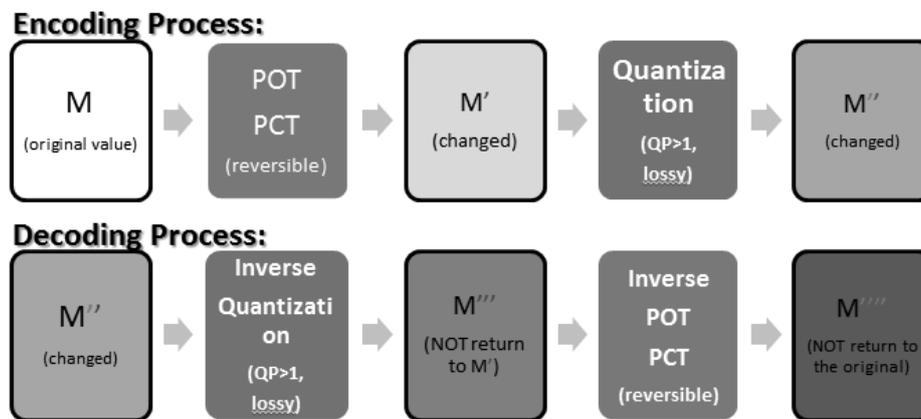


Figure 7. The “irreversibility” when QP > 1.

2.2. POT

The purpose of JPEG-XR’s POT is to decrease block artefacts caused by PCT. As such, its application area must interleave with that of PCT [18], as shown in Figure 8.

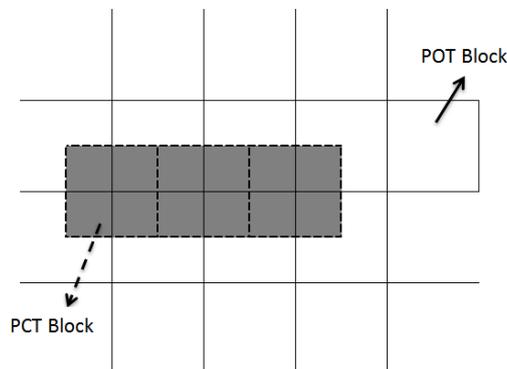


Figure 8. POT and PCT blocks interleave.

Since POT must interleave with PCT when undergoing a POT, the image must be divided into two regions. To each of the two regions would be applied a different transform as shown in Figure 9a,

depicting the border with 4×1 POT, while Figure 9b portrays the 4×4 POT inside. The location of the chequerboard and border artefacts happens to be where the 4×4 POT and 4×1 were applied. Additionally, the corner artefacts are located at the 42×2 corners where POT was not applied, as shown in Figures 10 and 11.

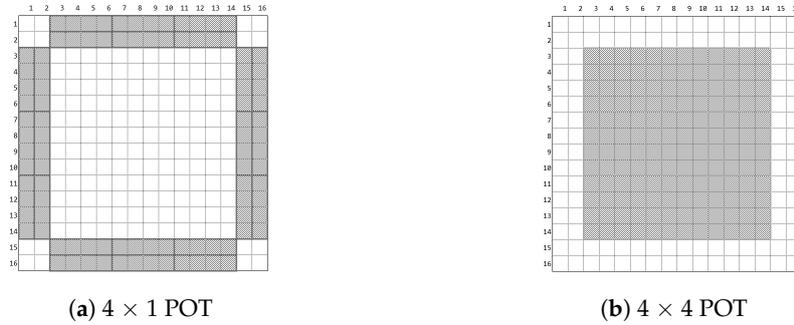


Figure 9. Two different types of POT application areas represented in grey.

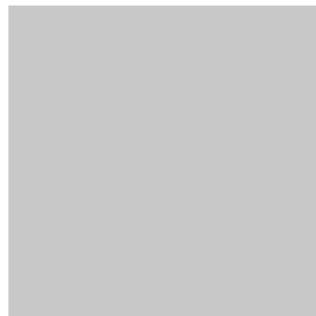
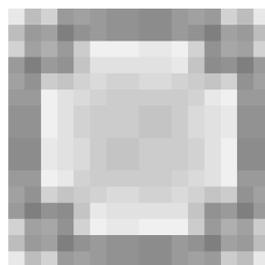
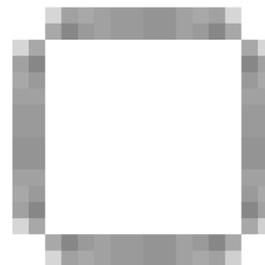


Figure 10. Original image, 16×16 grey BMP.



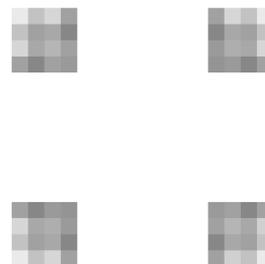
(a) QP = 61, with linear contrast enhancement



(b) Border artefacts, corresponding to a 4×1 area in Figure 9a



(c) Chequerboard artefacts, corresponding to a 4×4 area in Figure 9b.



(d) Chequerboard artefacts, corresponding to a 4×4 area in Figure 9b.

Figure 11. Chequerboard and border artefacts correspond to the two POT application areas, while the corner artefacts correspond to the four missing corners where POT was not applied.

Through research, we found the causes of the chequerboard, border and corner artefacts:

1. The 4×4 POT causes chequerboard artefacts
2. The 4×1 POT causes border artefacts
3. The four corners without POT application causes corner artefacts

Next, we will describe each of the three causes in detail.

2.3. The 4×4 POT: Chequerboard Block Artefacts

Based on [24], we know that the operations of 4×4 POT are separated into four stages, which by using the 16 points, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o and p, are given as:

1. Hadamard transform stage:

- (a) $T_{HH}(a, d, m, p)$
- (b) $T_{HH}(b, c, n, o)$
- (c) $T_{HH}(e, h, i, l)$
- (d) $T_{HH}(f, g, j, k)$

2. Scaling stage:

- (a) $T_S(a, p)$
- (b) $T_S(b, l)$
- (c) $T_S(e, o)$
- (d) $T_S(f, k)$

3. Rotation stage:

- (a) $T_R(n, m)$
- (b) $T_R(j, i)$
- (c) $T_R(h, d)$
- (d) $T_R(g, c)$
- (e) $T_{RR}(k, l, o, p)$

4. Hadamard transform stage:

- (a) $T_{HH}(a, d, m, p)$
- (b) $T_{HH}(b, c, n, o)$
- (c) $T_{HH}(e, h, i, l)$
- (d) $T_{HH}(f, g, j, k)$

Each of the above stages uses four rotational operators: T_R , T_{HH} , T_{RR} and T_S . Figure 12a–d shows the implementations of T_R , T_{HH} , T_{RR} and T_S in DPK [26], respectively.

From [18], we know that the four rotational operators listed in Figure 12 are implemented by the lifting structure [24,25,27]. Its basic structure is as shown in Figure 13. The advantage of using the lifting structure is that the transform operations can be separated into simple addition, subtraction and multiplication (e.g., Figure 13a can be expanded into Equation (1)). Moreover, inverse transform operations can be achieved by simply reversing the operation order, and switching the pluses and minuses (e.g., Equation (2) is the result of expanding Figure 13b).

$$\begin{aligned} x' &= x + a^*y \\ y' &= y + b^*x' \end{aligned} \tag{1}$$

$$\begin{aligned}
 y &= y' - b^* x' \\
 x &= x' - a^* y
 \end{aligned}
 \tag{2}$$

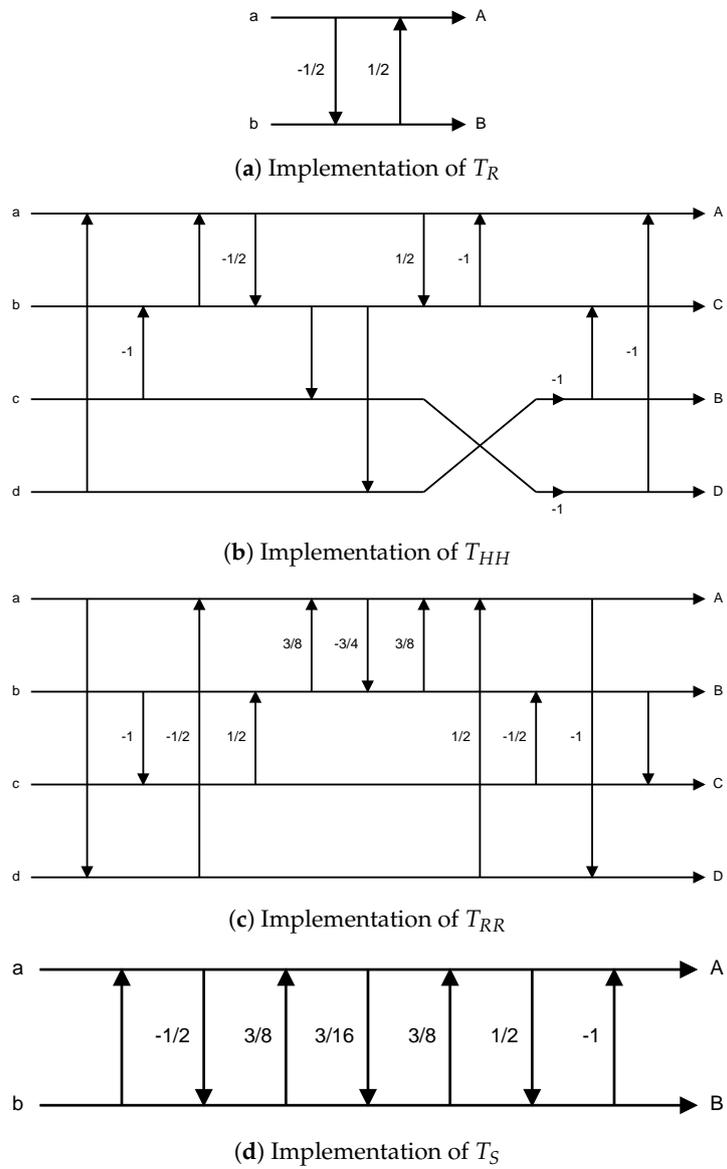


Figure 12. Implementations of JPEG XR POT rotation operators in Microsoft DPK.

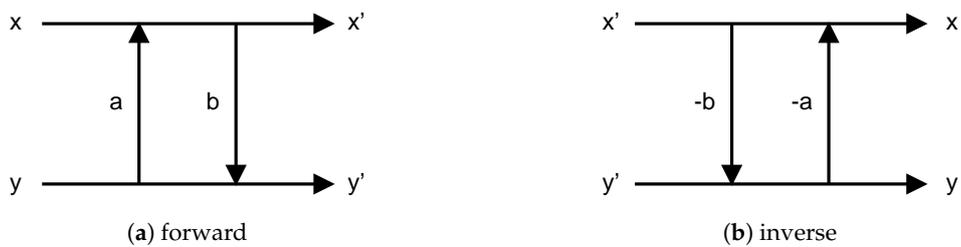


Figure 13. Lifting structure.

Using the method of Equation (1) and Figure 13a, we substitute from Figure 12a–d for the four POT rotational operators— T_R , T_{HH} , T_{RR} , T_S —and expand each. The results are shown in Equations (3)–(6).

$$T_R(a, b) = \begin{bmatrix} 3/4 & 1/2 \\ -1/2 & 1 \end{bmatrix} \bullet \begin{bmatrix} a \\ b \end{bmatrix} \quad (3)$$

$$T_{HH}(a, b, c, d) = 1/2 \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \bullet \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \quad (4)$$

$$T_{RR}(a, b, c, d) = 1/512 \begin{bmatrix} 440 & 165 & 165 & 72 \\ -192 & 440 & -72 & 192 \\ -192 & -72 & 440 & 192 \\ 72 & -165 & -165 & 440 \end{bmatrix} \bullet \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \quad (5)$$

$$T_S(a, b) = 1/4096 \begin{bmatrix} 2821 & 27 \\ -27 & 5947 \end{bmatrix} \bullet \begin{bmatrix} a \\ b \end{bmatrix} \quad (6)$$

The expanded results as shown in Equations (7) and (8) were obtained.

$$f_{POT4 \times 4} \left(\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \right) = \begin{bmatrix} \vec{C}_A \bullet \vec{V} & \vec{C}_B \bullet \vec{V} & \vec{C}_C \bullet \vec{V} & \vec{C}_D \bullet \vec{V} \\ \vec{C}_E \bullet \vec{V} & \vec{C}_F \bullet \vec{V} & \vec{C}_G \bullet \vec{V} & \vec{C}_H \bullet \vec{V} \\ \vec{C}_I \bullet \vec{V} & \vec{C}_J \bullet \vec{V} & \vec{C}_K \bullet \vec{V} & \vec{C}_L \bullet \vec{V} \\ \vec{C}_M \bullet \vec{V} & \vec{C}_N \bullet \vec{V} & \vec{C}_O \bullet \vec{V} & \vec{C}_P \bullet \vec{V} \end{bmatrix} \quad (7)$$

Among them:

$$\begin{aligned} \vec{V} &= [a \ b \ c \ d \ e \ f \ \dots \ l \ m \ n \ o \ p]^T \\ \vec{C}_A &= [0.9843 \ -0.2414 \ 0.2425 \ \dots \ -0.007494 \ 0.008556 \ -0.01566] \\ \vec{C}_B &= [0.2605 \ 0.9218 \ -0.08032 \ \dots \ -0.2053 \ 0.04684 \ 0.0105] \\ \vec{C}_C &= [-0.2605 \ -0.07419 \ 0.9214 \ \dots \ 0.04638 \ -0.1992 \ -0.0105] \\ \vec{C}_D &= [-0.1367 \ 0.2414 \ -0.2425 \ \dots \ 0.007494 \ -0.008556 \ -0.1367] \\ \vec{C}_E &= [0.2605 \ -0.05081 \ 0.05128 \ \dots \ 0.05128 \ -0.05081 \ 0.0105] \\ \vec{C}_F &= [0.05081 \ 0.2414 \ 0.007494 \ \dots \ -0.2425 \ -0.008556 \ 0.05081] \\ \vec{C}_G &= [-0.05081 \ 0.008556 \ 0.2425 \ \dots \ -0.007494 \ -0.2414 \ -0.05081] \\ \vec{C}_H &= [-0.0105 \ 0.05081 \ -0.05128 \ \dots \ -0.05128 \ 0.05081 \ -0.2605] \\ \vec{C}_I &= [-0.2605 \ 0.05081 \ -0.05128 \ \dots \ -0.05128 \ 0.05081 \ -0.0105] \\ \vec{C}_J &= [-0.05081 \ -0.2414 \ -0.007494 \ \dots \ 0.2425 \ 0.008556 \ -0.05081] \\ \vec{C}_K &= [0.05081 \ -0.008556 \ -0.2425 \ \dots \ 0.007494 \ 0.2414 \ 0.05081] \\ \vec{C}_L &= [0.0105 \ -0.05081 \ 0.05128 \ \dots \ 0.05128 \ -0.05081 \ 0.2605] \\ \vec{C}_M &= [-0.1367 \ -0.008556 \ 0.007494 \ \dots \ -0.2425 \ 0.2414 \ -0.1367] \\ \vec{C}_N &= [-0.0105 \ -0.1992 \ 0.04638 \ \dots \ 0.9214 \ -0.07419 \ -0.2605] \\ \vec{C}_O &= [0.0105 \ 0.04684 \ -0.2053 \ \dots \ -0.08032 \ 0.9218 \ 0.2605] \\ \vec{C}_P &= [-0.01566 \ 0.008556 \ -0.007494 \ \dots \ 0.2425 \ -0.2414 \ 0.9843] \end{aligned} \quad (8)$$

When we substitute a to p all with the same variable x , the result is Equation (9):

$$\begin{aligned}
 & f_{POT4 \times 4} \left(\begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \right) \\
 &= \begin{bmatrix} 0.686^*x & 0.684^*x & 0.693^*x & 0.691^*x \\ 0.684^*x & 0.678^*x & 0.7^*x & 0.693^*x \\ 0.693^*x & 0.7^*x & 0.678^*x & 0.684^*x \\ 0.691^*x & 0.693^*x & 0.684^*x & 0.686^*x \end{bmatrix}
 \end{aligned} \tag{9}$$

Using the same method as Equation (2) and Figure 13b, the inverse 4×4 POT can be expanded; the result is Equation (10).

$$\begin{aligned}
 & f_{IPOT4 \times 4} \left(\begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \right) \\
 &= \begin{bmatrix} 1.458^*x & 1.458^*x & 1.445^*x & 1.445^*x \\ 1.458^*x & 1.458^*x & 1.445^*x & 1.445^*x \\ 1.445^*x & 1.445^*x & 1.458^*x & 1.458^*x \\ 1.445^*x & 1.445^*x & 1.458^*x & 1.458^*x \end{bmatrix}
 \end{aligned} \tag{10}$$

A look at Equations (9) and (10) shows that both 4×4 POT and inverse 4×4 POT resulted in a chequerboard-like distribution, as shown in Figure 14. The values are displayed in greyscale. Assuming $x > 0$, in the 4×4 POT (Figure 14a), the top left and bottom right corners have lower values (colour is dark) while the top right and bottom left corners have higher values (colour is lighter). The inverse 4×4 POT, on the other hand, resulted in exactly the opposite chequerboard colours. The mirrored results appear in Figure 14b.

0.686^*x	0.684^*x	0.693^*x	0.691^*x
0.684^*x	0.678^*x	0.7^*x	0.693^*x
0.693^*x	0.7^*x	0.678^*x	0.684^*x
0.691^*x	0.693^*x	0.684^*x	0.686^*x

(a) 4×4 POT

1.458^*x	1.458^*x	1.445^*x	1.445^*x
1.458^*x	1.458^*x	1.445^*x	1.445^*x
1.445^*x	1.445^*x	1.458^*x	1.458^*x
1.445^*x	1.445^*x	1.458^*x	1.458^*x

(b) Inverse 4×4 POT

Figure 14. Chequerboard artefact distribution of 4×4 POT and inverse 4×4 POT.

To further understand how the chequerboard phenomenon occurred, we followed the encoding process described in Figure 4. First, the 4×4 PCT (represented as $f_{PCT4 \times 4}$) was applied to the 4×4 POT result from Equation (9). Note that since application areas of POT and PCT interleave (Section 2.2), the input array was relatively displaced. The operation result is shown in Equation (11).

$$\begin{aligned}
 & f_{PCT4 \times 4} \left(\begin{bmatrix} 0.678*x & 0.684*x & 0.693*x & 0.7*x \\ 0.684*x & 0.686*x & 0.691*x & 0.693*x \\ 0.693*x & 0.691*x & 0.686*x & 0.684*x \\ 0.7*x & 0.693*x & 0.684*x & 0.678*x \end{bmatrix} \right) \\
 &= \begin{bmatrix} 2.75*x - 0.5 & 0.5 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & -0.001*x & 0.0004*x \\ 0 & 0 & 0.0004*x & -0.0254*x \end{bmatrix}
 \end{aligned} \tag{11}$$

We know that PCT and DCT are similar [24]. An image is transformed from the spatial domain to the frequency domain. The distribution of transformed data depends on its frequency. For example, in Equation (11), the top left corner is Low-Pass (LP), while the others are High-Pass (HP). In Equation (11), the value distribution was checkerboard-like due to the previous 4×4 POT. After the 4×4 PCT, some weak signals were generated in the HP areas. Still, most of the signals were concentrated in the LP area, or the top left of Equation (11) ($2.75*x - 0.5$).

From [24], we know that JPEG-XR has a two-stage PCT, as shown in Figure 15. Each 4×4 block processes the first-stage PCT, resulting in an LP coefficient (light-grey) and 15 HP coefficients (white). Then, the second-stage transform was applied to the 16 DC coefficients collected into a single 4×4 block. These yielded 16 new coefficients, referred to the LP coefficient of the original block, respectively.

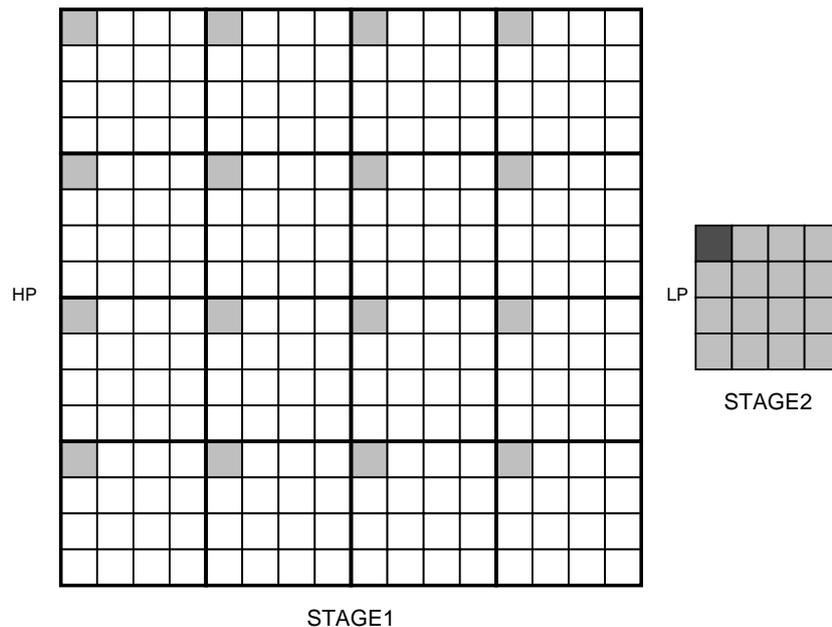


Figure 15. Two-stage PCT in JPEG-XR.

We now apply the second-stage 4×4 PCT to a 4×4 matrix, filling the LP value of the first-stage transform from Equation (11). The result is shown in Equation (12).

$$\begin{aligned}
 & f_{PCT4 \times 4} \left(\begin{bmatrix} 2.75*x - 0.5 & 2.75*x - 0.5 & 2.75*x - 0.5 & 2.75*x - 0.5 \\ 2.75*x - 0.5 & 2.75*x - 0.5 & 2.75*x - 0.5 & 2.75*x - 0.5 \\ 2.75*x - 0.5 & 2.75*x - 0.5 & 2.75*x - 0.5 & 2.75*x - 0.5 \\ 2.75*x - 0.5 & 2.75*x - 0.5 & 2.75*x - 0.5 & 2.75*x - 0.5 \end{bmatrix} \right) \\
 &= \begin{bmatrix} 11*x - 2.5 & 0.5 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
 \end{aligned} \tag{12}$$

In Section 2.1, we learned that when $QP > 1$, values became distorted after the process of quantization and inverse quantization. We set LP_{S2} instead of the LP value in Equation (11), which has been processed with the second-stage 4×4 PCT, $C_0 = 11^*x - 2.5$. We then used $QP = 61$ to perform quantization and inverse quantization for Equations (11) and (12) (shown as $f_{Q.IQ}()$). The results are Equations (13) and (14), respectively. From the source code of DPK [26], we know that under 8-bit greyscale conditions, after undergoing linear adjustments (offset) with the DPK program, the result is $|x| \leq 2048$. From Figure 3, we can see that when $QP = 61$ and $|c| > 144$, $c \in R$, then $f_{Q.IQ}(c) > 0$. This means if we would like $f_{Q.IQ}(k^*x) > 0$, $k \in R$, then $|k| \geq 144/2048$, or $|k| \geq 0.0703125$. From this, we learn that in the HP area of Equations (11) and (12), the values were adjusted to zero after the $f_{Q.IQ}()$ calculation, as they were too small ($< 0.0703125^*x$). At the end, only the LP portion, or the $f_{Q.IQ}(C_0)$, of the top left corner remained, as shown in Equation (14).

$$f_{Q.IQ}\left(\begin{bmatrix} LP_{S2} & 0.5 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & -0.001^*x & 0.0004^*x \\ 0 & 0 & 0.0004^*x & -0.0254^*x \end{bmatrix}\right) = \begin{bmatrix} f_{Q.IQ}(LP_{S2}) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (13)$$

$$f_{Q.IQ}\left(\begin{bmatrix} C_0 & 0.5 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\right) = \begin{bmatrix} f_{Q.IQ}(C_0) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (14)$$

Next, we set $C_1 = f_{Q.IQ}(C_0)$ and performed inverse second-stage PCT 4×4 transform on Equation (14) (shown as $f_{IPCT4 \times 4}$). The result is Equation (15). One can observe that besides the four corners being $0.25^*(C_1 - 1)$, all other points were equivalent to $0.25^*(C_1 + 1)$.

$$f_{IPCT4 \times 4}\left(\begin{bmatrix} C_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\right) = 0.25^*\left(\begin{bmatrix} C_1 - 1 & C_1 + 1 & C_1 + 1 & C_1 - 1 \\ C_1 + 1 & C_1 + 1 & C_1 + 1 & C_1 + 1 \\ C_1 + 1 & C_1 + 1 & C_1 + 1 & C_1 + 1 \\ C_1 - 1 & C_1 + 1 & C_1 + 1 & C_1 - 1 \end{bmatrix}\right) \quad (15)$$

From Figure 3, we know that when $QP = 61$, if $|f_{Q.IQ}(c)| > 0$, then $|f_{Q.IQ}(c)| \geq 232$, and $C_1 = f_{Q.IQ}(C_0)$, or $|C_1| \geq 232$; therefore, C_1 is large enough to make $C_1 + 1 \approx C_1$ and $C_1 - 1 \approx C_1$. Then, we can set $LP_{S2} = 0.25^*C_1 = C_2$ and perform the inverse first-stage PCT 4×4 transform on Equation (13). The result is Equation (16):

$$\begin{aligned}
 & f_{IPCT4 \times 4} \left(\begin{bmatrix} C_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \right) \\
 &= 0.25^* \left(\begin{bmatrix} C_2 - 1 & C_2 + 1 & C_2 + 1 & C_2 - 1 \\ C_2 + 1 & C_2 + 1 & C_2 + 1 & C_2 + 1 \\ C_2 + 1 & C_2 + 1 & C_2 + 1 & C_2 + 1 \\ C_2 - 1 & C_2 + 1 & C_2 + 1 & C_2 - 1 \end{bmatrix} \right)
 \end{aligned} \tag{16}$$

Finally, we perform inverse POT on Equation (16) (shown as $f_{IPOT4 \times 4}$). The result is Equation (17).

$$\begin{aligned}
 & f_{IPOT4 \times 4} \left(0.25^* \left(\begin{bmatrix} C_2 - 1 & C_2 + 1 & C_2 + 1 & C_2 - 1 \\ C_2 + 1 & C_2 + 1 & C_2 + 1 & C_2 + 1 \\ C_2 + 1 & C_2 + 1 & C_2 + 1 & C_2 + 1 \\ C_2 - 1 & C_2 + 1 & C_2 + 1 & C_2 - 1 \end{bmatrix} \right) \right) \\
 &= \begin{bmatrix} 0.365^*(C_2 - 1) & 0.365^*(C_2 + 1) & 0.361^*(C_2 + 1) & 0.361^*(C_2 - 1) \\ 0.365^*(C_2 + 1) & 0.365^*(C_2 + 1) & 0.361^*(C_2 + 1) & 0.361^*(C_2 + 1) \\ 0.361^*(C_2 + 1) & 0.361^*(C_2 + 1) & 0.365^*(C_2 + 1) & 0.365^*(C_2 + 1) \\ 0.361^*(C_2 - 1) & 0.361^*(C_2 + 1) & 0.365^*(C_2 + 1) & 0.365^*(C_2 - 1) \end{bmatrix}
 \end{aligned} \tag{17}$$

The final result of Equation (17) can be rearranged into Figure 16. Assuming $C_2 > 0$ with greyscale representation, the top left and lower right corners have higher values (light grey), while the top right and lower left corners have lower values (dark grey). This distribution is like a chequerboard. Up to now, we have explained the cause for chequerboard artefacts: lossy quantization caused the HP signals to be cleaned out to zero, leaving only the LP signals (Equation (13)). Then, after the inverse 4×4 PCT, it can be observed that the results were evenly distributed, or all 16 values were pretty much equal (Equation (16)). At last, the inverse 4×4 POT was applied, and a chequerboard-like phenomenon similar to Equation (10) can be seen.

0.365^*C_2	0.365^*C_2	0.361^*C_2	0.361^*C_2
0.365^*C_2	0.365^*C_2	0.361^*C_2	0.361^*C_2
0.361^*C_2	0.361^*C_2	0.365^*C_2	0.365^*C_2
0.361^*C_2	0.361^*C_2	0.365^*C_2	0.365^*C_2

Figure 16. The 4×4 region after encoding and decoding; the distribution is chequerboard-like.

2.4. The 4×1 POT: Border Artefacts

The implementation of the 4×1 POT using the DPK program [26] is shown in Figure 17.

Referring to the method of Equation (1) and Figure 13a, we used a , b , c and d to represent the four points, then substituted them into Figure 17 and expanded. The result is Equation (18)

$$f_{POT4 \times 1} \left(\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \right) = \begin{bmatrix} 1.02^*a - 0.301^*b + 0.302^*c - 0.189^*d \\ 0.301^*a + 0.867^*b - 0.0383^*c - 0.302^*d \\ -0.301^*a - 0.0369^*b + 0.867^*c + 0.302^*d \\ -0.188^*a + 0.301^*b - 0.302^*c + 1.02^*d \end{bmatrix} \tag{18}$$

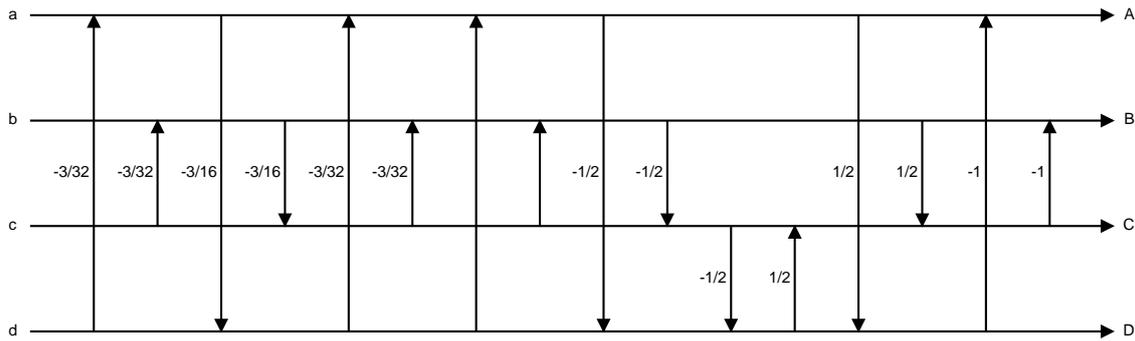


Figure 17. Implementation of 4×1 POT.

Similarly, we made $a = b = c = d = x$, and the result after substitution is Equation (19).

$$f_{POT4 \times 1} \begin{pmatrix} x \\ x \\ x \\ x \end{pmatrix} = \begin{bmatrix} 0.8288 * x \\ 0.8282 * x \\ 0.8303 * x \\ 0.8297 * x \end{bmatrix} \tag{19}$$

Similarly, we can expand the calculation for the inverse 4×1 POT where the result is Equation (20).

$$f_{IPOT4 \times 1} \begin{pmatrix} x \\ x \\ x \\ x \end{pmatrix} = \begin{bmatrix} 1.207 * x \\ 1.207 * x \\ 1.205 * x \\ 1.205 * x \end{bmatrix} \tag{20}$$

We combined the results of Equations (9), (10), (19) and (20) according to the upper boundary application area of Figure 9. The result is shown in Figure 18. The top two rows are the area where 4×1 POT was applied. The bottom four rows are where 4×4 POT was applied. One can see that in Figure 18, the value of 4×1 POT is bigger than the lower 4×4 POT (light grey). In Figure 18b, the value of inverse 4×1 POT is less than the lower inverse 4×4 POT (dark grey).

0.8288*x	0.8282*x	0.8303*x	0.8297*x	← 4x1 POT
0.8288*x	0.8282*x	0.8303*x	0.8297*x	
0.686*x	0.684*x	0.693*x	0.691*x	← 4x4 POT
0.684*x	0.678*x	0.7*x	0.693*x	
0.691*x	0.7*x	0.678*x	0.684*x	
0.693*x	0.693*x	0.684*x	0.686*x	

(a) 4×1 POT and 4×4 POT

1.207*x	1.207*x	1.205*x	1.205*x	← Inverse-4x1 POT
1.207*x	1.207*x	1.205*x	1.205*x	
1.458*x	1.458*x	1.445*x	1.445*x	← Inverse-4x4 POT
1.458*x	1.458*x	1.445*x	1.445*x	
1.445*x	1.445*x	1.458*x	1.458*x	
1.445*x	1.445*x	1.458*x	1.458*x	

(b) Inverse 4×1 POT and inverse 4×4 POT

Figure 18. Border artefact distribution of 4×1 POT and inverse 4×1 POT.

Next, let us attempt to understand how border artefacts were formed. Here, we will also consider the interleaving between application areas of POT and PCT. Please refer to Figure 18. To represent the actual state of the application for the upper image boundary, we took two horizontal 4×1 POT areas for the top two rows. The next two rows are the top half of the 4×4 POT area. This is shown in Equation (21).

$$\begin{bmatrix} 0.8303*x & 0.8297*x & 0.8288*x & 0.8282*x \\ 0.8303*x & 0.8297*x & 0.8288*x & 0.8282*x \\ 0.693*x & 0.691*x & 0.686*x & 0.684*x \\ 0.7*x & 0.693*x & 0.684*x & 0.678*x \end{bmatrix} \quad (21)$$

Using the same steps as Section 2.3, we applied PCT, quantization, inverse PCT and inverse POT operations on Equation (21). The result is Equation (22), where $C_{4 \times 1} = f_{Q.IQ}(3.04*x - 0.5)$ and $C_{4 \times 4} = f_{Q.IQ}(2.75*x - 0.5)$:

$$\begin{bmatrix} 0.301*C_{4 \times 1} & 0.301*C_{4 \times 1} & 0.302*C_{4 \times 1} & 0.302*C_{4 \times 1} \\ 0.301*C_{4 \times 1} & 0.301*C_{4 \times 1} & 0.302*C_{4 \times 1} & 0.302*C_{4 \times 1} \\ 0.337*C_{4 \times 1} & 0.337*C_{4 \times 1} & 0.339*C_{4 \times 1} & 0.339*C_{4 \times 1} \\ +0.0244*C_{4 \times 4} & +0.0244*C_{4 \times 4} & +0.0261*C_{4 \times 4} & +0.0261*C_{4 \times 4} \\ 0.243*C_{4 \times 1} & 0.243*C_{4 \times 1} & 0.245*C_{4 \times 1} & 0.245*C_{4 \times 1} \\ +0.118*C_{4 \times 4} & +0.118*C_{4 \times 4} & +0.12*C_{4 \times 4} & +0.12*C_{4 \times 4} \end{bmatrix} \quad (22)$$

Considering the characteristics of quantization, when $x > 0$, $C_{4 \times 1} \geq C_{4 \times 4}$. This means we can make $C_{4 \times 1} = C_{4 \times 4} + K$ where $K \geq 0$. Therefore, the result of Equation (22) can be reordered into Figure 19. Assuming $C_{4 \times 1} > 0$, using greyscale representation, one can see that the top two rows (where 4×1 POT was applied) have lower values (dark grey). We only took half (4×2) of the bottom two rows (where 4×4 POT was applied), so only half of the chequerboard is shown here, but all the values in this area were higher than the top two rows (lighter grey). Up to now, we have explained the cause of border artefacts: this is due to different transformation values resulting from inverse 4×1 POT and inverse 4×4 POT.

$0.301*C_{4 \times 1}$	$0.301*C_{4 \times 1}$	$0.302*C_{4 \times 1}$	$0.302*C_{4 \times 1}$
$0.301*C_{4 \times 1}$	$0.301*C_{4 \times 1}$	$0.302*C_{4 \times 1}$	$0.302*C_{4 \times 1}$
$0.3614*C_{4 \times 1}$	$0.3614*C_{4 \times 1}$	$0.3651*C_{4 \times 1}$	$0.3651*C_{4 \times 1}$
$+0.0244*K$	$+0.0244*K$	$+0.0261*K$	$+0.0261*K$
$0.361*C_{4 \times 1}$	$0.361*C_{4 \times 1}$	$0.365*C_{4 \times 1}$	$0.365*C_{4 \times 1}$
$+0.118*K$	$+0.118*K$	$+0.12*K$	$+0.12*K$

Figure 19. The 4×1 region after encoding and decoding, causing border artefacts.

2.5. Corner Artefacts

Combining Sections 2.3 and 2.4, we simulated the top left 4×4 area of the image, i.e., not applying POT to the top left 2×2 corner, applying 4×1 POT to the top right and bottom left corners and applying 4×4 POT to the bottom right corner. The result is Equation (23).

$$\begin{bmatrix} x & x & 0.8288*x & 0.8282*x \\ x & x & 0.8288*x & 0.8282*x \\ 0.8288*x & 0.8288*x & 0.686*x & 0.684*x \\ 0.8282*x & 0.8282*x & 0.684*x & 0.678*x \end{bmatrix} \quad (23)$$

We applied the aforementioned 4×4 PCT on Equation (23). The result is Equation (24).

$$\begin{aligned}
 & f_{PCT4 \times 4} \left(\begin{bmatrix} x & x & 0.8288^*x & 0.8282^*x \\ x & x & 0.8288^*x & 0.8282^*x \\ 0.8288^*x & 0.8288^*x & 0.686^*x & 0.684^*x \\ 0.8282^*x & 0.8282^*x & 0.684^*x & 0.678^*x \end{bmatrix} \right) \\
 &= \begin{bmatrix} 3.34^*x - 0.5 & 0.5 - 0.002433^*x & 0.2733^*x & 0.002163^*x \\ 0.5 - 0.002433^*x & 0.5 - 0.000927^*x & 0.0001159^*x & -0.1164^*x \\ 0.2733^*x & 0.0001159^*x & 0.005162^*x & -0.01103^*x \\ 0.002163^*x & -0.1164^*x & -0.01103^*x & 0.0199^*x \end{bmatrix}
 \end{aligned} \tag{24}$$

One can see from Equation (24) that the magnitude of the value change in the top left corner is greater because we did not apply POT to it. Therefore, after undergoing PCT, more signals would be generated in the HP area. Next, we also applied quantization and inverse quantization to Equation (24) (represented by $f_{Q,IQ}()$). The result is Equation (25).

$$\begin{aligned}
 & f_{Q,IQ} \left(\begin{bmatrix} 3.34^*x - 0.5 & 0.5 - 0.002433^*x & 0.2733^*x & 0.002163^*x \\ 0.5 - 0.002433^*x & 0.5 - 0.000927^*x & 0.0001159^*x & -0.1164^*x \\ 0.2733^*x & 0.0001159^*x & 0.005162^*x & -0.01103^*x \\ 0.002163^*x & -0.1164^*x & -0.01103^*x & 0.0199^*x \end{bmatrix} \right) \\
 &= \begin{bmatrix} f_{Q,IQ}(3.34^*x - 0.5) & 0 & 0 & 0 \\ 0 & 0 & 0 & f_{Q,IQ}(-0.1164^*x) \\ f_{Q,IQ}(0.2733^*x) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
 \end{aligned} \tag{25}$$

Note that the result of Equation (25) is different from Sections 2.3 and 2.4. The HP signals were not completely transformed to zero. Since $f_{Q,IQ}(0.2733^*x)$ and $f_{Q,IQ}(-0.1164^*x)$ satisfied the aforementioned $|k| \geq 0.0703125$, they were kept. The decoding result differed from Sections 2.3 and 2.4 and is the reason for corner artefacts. Up to now, we have explained the cause of corner artefacts.

3. Improvement

From the analysis in Section 2, we discovered three problems in POT: (1) 4×4 POT is “uneven” (different values/non-uniform), which causes chequerboard artefacts; (2) the transform result value of 4×1 POT is greater than 4×4 POT, which causes border artefacts; and (3) POT was not applied on 2×2 areas in the four corners, which causes corner artefacts. To solve these problems, we propose the use of a “uniform matrix”. After the POT (encoding) step and before the inverse POT (decoding) step, our uniform matrix was multiplied to even things out. After applying our solution, the process is as described in Figure 20.

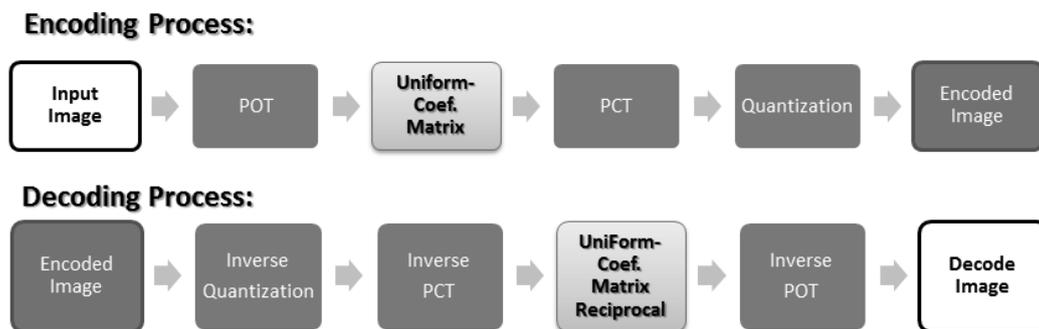


Figure 20. Improved JPEG-XR encoding/decoding process.

3.1. Improvement of 4×4 POT

As shown in Equation (9), the result of 4×4 POT is “uneven”. We hope to obtain a uniform matrix so that the Hadamard product between it and 4×4 POT becomes even. We took the average value of 16 points (0.6887^*x) that have undergone the 4×4 POT as our evenness:

$$\begin{aligned}
 & \begin{bmatrix} 0.686^*x & 0.684^*x & 0.693^*x & 0.691^*x \\ 0.684^*x & 0.678^*x & 0.7^*x & 0.693^*x \\ 0.693^*x & 0.7^*x & 0.678^*x & 0.684^*x \\ 0.691^*x & 0.693^*x & 0.684^*x & 0.686^*x \end{bmatrix} \odot [\text{uniform matrix}] \\
 & = \begin{bmatrix} 0.6887^*x & 0.6887^*x & 0.6887^*x & 0.6887^*x \\ 0.6887^*x & 0.6887^*x & 0.6887^*x & 0.6887^*x \\ 0.6887^*x & 0.6887^*x & 0.6887^*x & 0.6887^*x \\ 0.6887^*x & 0.6887^*x & 0.6887^*x & 0.6887^*x \end{bmatrix}
 \end{aligned} \tag{26}$$

After calculations, the uniform matrix of 4×4 POT is:

$$\begin{bmatrix} 1.004 & 1.007 & 0.9936 & 0.997 \\ 1.007 & 1.016 & 0.9849 & 0.9936 \\ 0.9936 & 0.9849 & 1.016 & 1.007 \\ 0.997 & 0.9936 & 1.007 & 1.004 \end{bmatrix} \tag{27}$$

The corresponding inverse uniform matrix is its reciprocal.

$$\begin{bmatrix} 1/1.004 & 1/1.007 & 1/0.9936 & 1/0.997 \\ 1/1.007 & 1/1.016 & 1/0.9849 & 1/0.9936 \\ 1/0.9936 & 1/0.9849 & 1/1.016 & 1/1.007 \\ 1/0.997 & 1/0.9936 & 1/1.007 & 1/1.004 \end{bmatrix} \tag{28}$$

For the ease of implementation, we adjusted the uniform matrix into a fraction. We set the numerator to 256 (2^8) and took an approximate value. This way, we can implement using the integer data structure and bit shifting in the program in order to maintain accuracy and decrease complexity. Finally, we get Equations (29) and (30), which are respectively the uniform matrix and inverse uniform matrix of 4×4 POT.

$$256/ \begin{bmatrix} 255 & 254 & 258 & 257 \\ 254 & 252 & 260 & 258 \\ 258 & 260 & 252 & 254 \\ 257 & 258 & 254 & 255 \end{bmatrix} \tag{29}$$

$$\begin{bmatrix} 255 & 254 & 258 & 257 \\ 254 & 252 & 260 & 258 \\ 258 & 260 & 252 & 254 \\ 257 & 258 & 254 & 255 \end{bmatrix} / 256 \tag{30}$$

3.2. Improvement of 4×1 POT

Next, we will attempt to improve the result of 4×1 POT. Through [24], we discovered that 4×1 POT is a 1D transform, and 4×4 POT is a 2D transform. This means that 4×4 POT has one more dimension than 4×1 POT, which causes differences in the transformation result. We hope to modify the 4×1 POT to make its transformation result the same as the 4×4 POT. The easiest way is to apply 4×4 POT instead of 4×1 POT on the area, but the problem with this is that a 4×1 POT area has only four points, while 4×4 POT needs 16 points. By observing Equations (7) and (8), we found:

$$\begin{aligned}
 \vec{C}_A[i] &= \vec{C}_P[(16-i)] \\
 \vec{C}_B[i] &= \vec{C}_O[(16-i)] \\
 &\vdots \\
 \vec{C}_F[i] &= \vec{C}_I[(16-i)] \\
 \vec{C}_G[i] &= \vec{C}_H[(16-i)]
 \end{aligned}
 \tag{31}$$

By Equation (31), we proved that 4×4 POT is “symmetrical”, as shown in Equation (32). If 4×4 POT was performed on an input of $8 \times 2 = 16$ points consisting of eight symmetrical points a-h, the result would also be symmetrical (A-H).

$$f_{POT_{4 \times 4}} \left(\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ h & g & f & e \\ d & c & b & a \end{bmatrix} \right) = \begin{bmatrix} A & B & C & D \\ E & F & G & H \\ H & G & F & E \\ D & C & B & A \end{bmatrix}
 \tag{32}$$

Due to the symmetrical characteristic of 4×4 POT, we can combine the 4×1 POT and its neighbouring 4×1 POT area to form $4 \times 2 = 8$ points, then expand the area into 4×4 POT size through symmetrization (mirroring). Now, the 4×4 POT may be applied. The process is as shown in Figure 21. Similarly, we can take advantage of this characteristic when decoding to convert into inverse 4×4 POT. To maintain the evenness of the 4×4 POT results, it is necessary to multiply the uniform matrix mentioned in Section 3.1 (Equations (29) and (30)) to the process described in Figure 21.

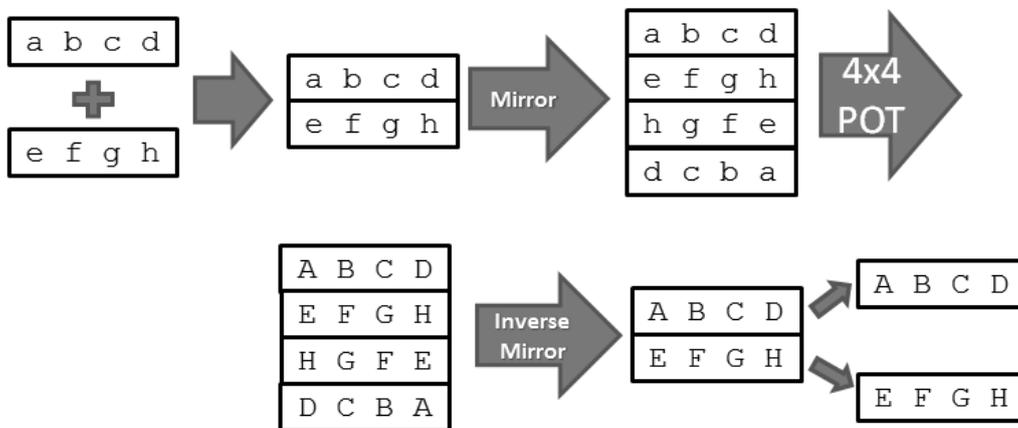


Figure 21. Using 4×4 POT instead of 4×1 POT.

Finally, we have the improvement function $f_{POT_{4 \times 2}}$. By making $a = b = c = d = e = f = g = h = x$, then substituting, this will yield the result described in Equation (33). Its value is $0.6887 * x$, which is equivalent to the improved 4×4 POT (Equation (26)).

$$\begin{aligned}
 &f_{POT_{4 \times 2}} \left(\begin{bmatrix} a & b & c & d \\ e & f & g & h \end{bmatrix} \right) \\
 &= \begin{bmatrix} 0.6887 * x & 0.6887 * x & 0.6887 * x & 0.6887 * x \\ 0.6887 * x & 0.6887 * x & 0.6887 * x & 0.6887 * x \end{bmatrix}
 \end{aligned}
 \tag{33}$$

3.3. Improvement of Corner Artefacts (2 × 2 POT)

From the POT application areas marked on Figure 9, we found that there was a 2 × 2 area in each of the four corners where POT was not applied. Their values were different than the areas where POT was applied. This scenario leads to block artefacts in the four corners. In order to ameliorate this phenomenon, we found the implementation of 2 × 2 POT in DPK, as shown in Figure 22.

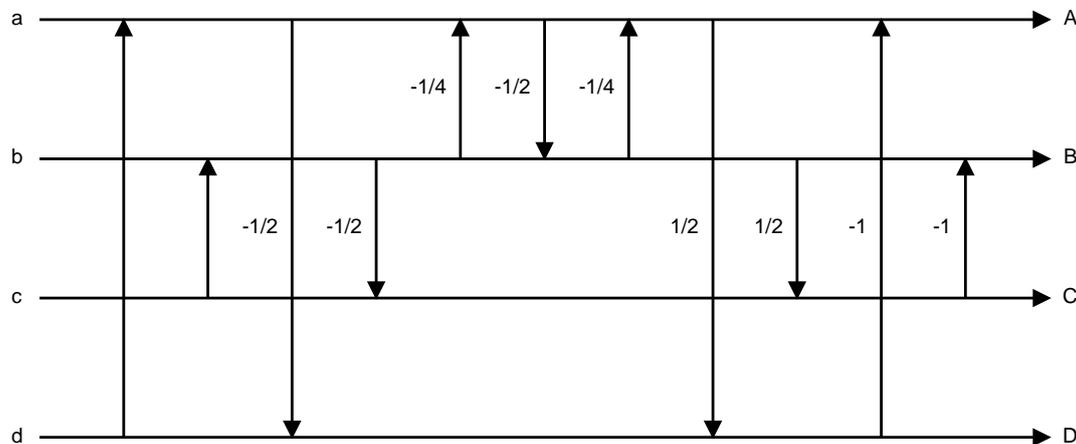


Figure 22. Implementation of 2 × 2 POT.

Here, we also applied the aforementioned expansion method; represent four points with a, b, c and d, substitute into Figure 22 and expand. The result is Equation (34):

$$f_{POT4 \times 2} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 1.062*a - 0.25*b - 0.25*c + 0.0625*d \\ -0.2656*a + 1.062*b + 0.0625*c - 0.2656*d \\ -0.2656*a + 0.0625*b + 1.062*c - 0.2656*d \\ 0.0625*a - 0.25*b - 0.25*c + 1.062*d \end{pmatrix} \quad (34)$$

We make $a = b = c = d = x$ and substitute to get:

$$f_{POT4 \times 2} \begin{pmatrix} x \\ x \\ x \\ x \end{pmatrix} = \begin{pmatrix} 0.625*x \\ 0.5938*x \\ 0.5938*x \\ 0.625*x \end{pmatrix} \quad (35)$$

Since we want the same result as that of the aforementioned improved results of 4 × 4 POT and 4 × 1 POT (Sections 3.1 and 3.2), here the target value was set to 0.6887*x, i.e.,

$$\begin{pmatrix} 0.625*x \\ 0.5938*x \\ 0.5938*x \\ 0.625*x \end{pmatrix} \odot [\text{uniform matrix of POT } 2 \times 2] = \begin{pmatrix} 0.6887*x \\ 0.6887*x \\ 0.6887*x \\ 0.6887*x \end{pmatrix} \quad (36)$$

The resulting uniform matrix of POT 2 × 2 is as shown below.

$$\begin{bmatrix} 1.102 & 1.16 & 1.16 & 1.102 \end{bmatrix}^T \quad (37)$$

Inverse uniform matrix:

$$\left[1/1.102 \quad 1/1.16 \quad 1/1.16 \quad 1/1.102\right]^T \quad (38)$$

Finally, we get Equations (39) and (40), which are the uniform matrix and inverse uniform matrix of the 2×2 POT in the four missing corners.

$$256/ \left[232 \quad 221 \quad 221 \quad 232\right]^T \quad (39)$$

$$\left[232 \quad 221 \quad 221 \quad 232\right]^T / 256 \quad (40)$$

4. Experimental Results

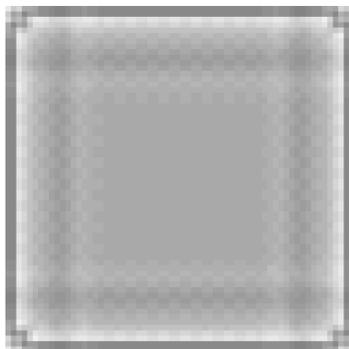
First, let us take a look at how a monochrome image has been improved. As shown in Figure 23, one can clearly see that chequerboard, border and corner artefacts have been eliminated.



(a) DPK
PSNR = 37.9396



(b) Proposed method
PSNR = 48.1648



(c) DPK, with linear
contrast enhancement



(d) Proposed method, with linear contrast enhancement

Figure 23. Testing with a monochrome image. After improvement, chequerboard, border and corner artefacts have been eliminated.

Next, we tested using the six images in Figures 24–26 (all are 8bpp greyscale images). We performed the JPEG-XR encoding and decoding process using two methods: Microsoft DPK [26] and our improvement program from Section 3. The overlap parameter was fixed at $L = 1$, while the QP was set from 21–101 with an interval of 20. Additionally, 11 were added to observe the performance under a low compression ratio. Calculated separately, PSNR and SSIM [28,29] were used to assess the coding efficiency. The results appear in Figures 27–31.



(a) White 1, 2560 × 3300

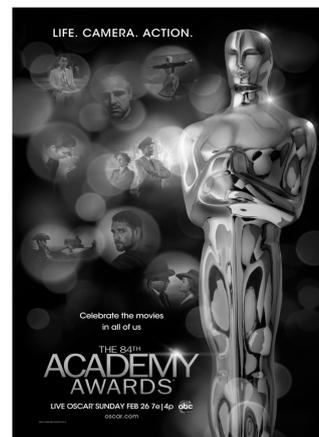


(b) White 2, 3300 × 2560

Figure 24. Test images, Part 1.



(a)



(b)

Figure 25. Test images, Part 2. (a) logo, 1024 × 768; (b) Poster [30], 840 × 1224.



(a)



(b)

Figure 26. Test images, Part 3. (a) Lena, 512 × 512; (b) sketch [31], 1080 × 1318.



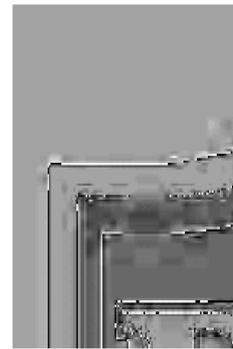
(a) DPK



(b) Proposed method

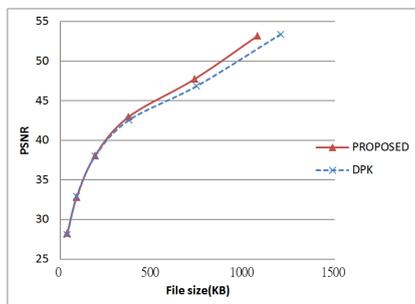


(c) DPK, with linear contrast enhancement.

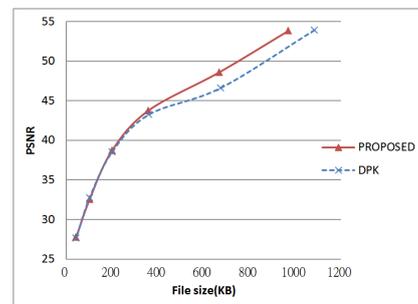


(d) Proposed method, with linear contrast enhancement.

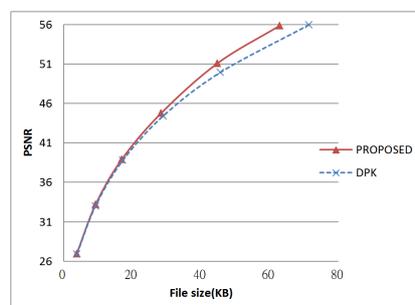
Figure 27. Visual comparison of the logo image (partial, 150×150).



(a) PSNR of White 1, 2560×3300

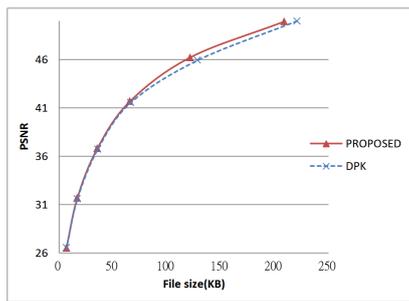


(b) PSNR of White 2, 3300×2560

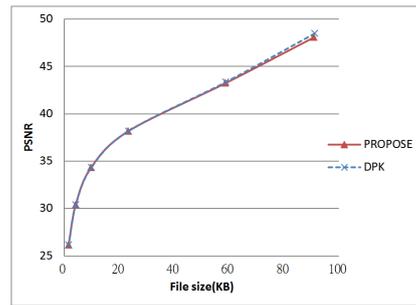


(c) PSNR of logo, 1024×768

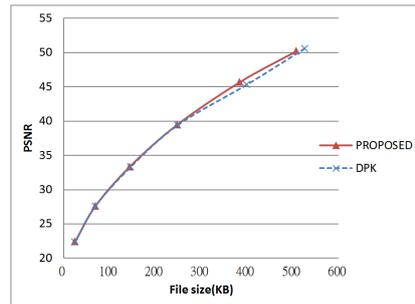
Figure 28. PSNR of test results, Part 1.



(a) PSNR of poster, 840×1224

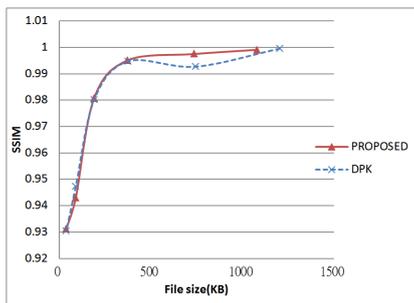


(b) PSNR of Lena, 512×512

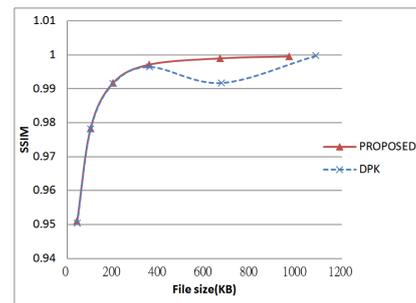


(c) PSNR of sketch [31], 1080×1318

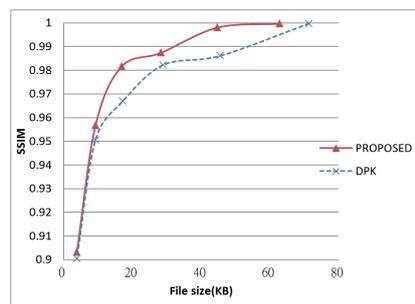
Figure 29. PSNR of test results, Part 2.



(a) SSIM of White 1, 2560×3300

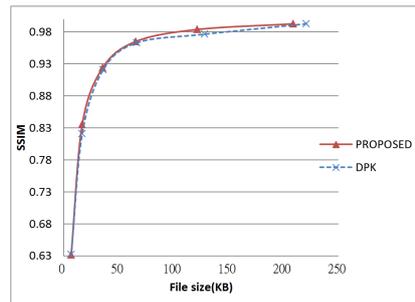
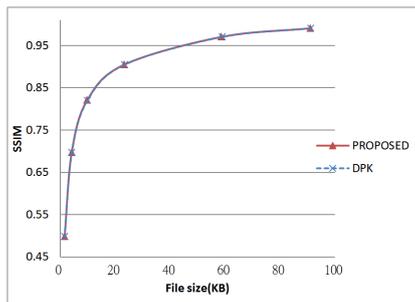
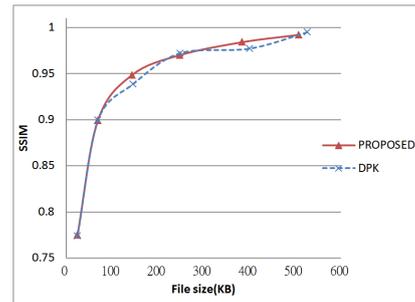


(b) SSIM of White 2, 3300×2560



(c) SSIM of logo, 1024×768

Figure 30. SSIM of test results, Part 1.

(a) SSIM of poster, 840×1224 (b) SSIM of Lena, 512×512 (c) SSIM of sketch [31], 1080×1318 **Figure 31.** SSIM of test results, Part 2.

Figures 28a,b and 30a,b are the results of Figure 24 respectively. These two images have wider white borders to simulate photo reproduction and scans. The results show that our improvement method yielded better performance when the compression ratio was medium to low ($QP \leq 41$). PSNR increased by as much as 1.97 dB, while SSIM increased by as much as 0.00727. Moreover, when the compression ratio was low ($QP \leq 11$), the resulting file size was as much as 10% smaller than DPK.

Figures 28c, 29a, 30c and 31a are the results of Figure 25 respectively. These two images are a common advertising logo and a poster. The results show that PSNR increased by as much as 1.14 dB, while SSIM increased by as much as 0.141. Moreover, when $QP = 11$, the file size decreased by 5%–10%.

Figures 29b and 31b are the result of Figure 26a using the “Lena” picture (512×512 pixel). Using our improvement method, the result was almost the same as the DPK when $QP \geq 50$. The reason there was little room for improvement was that checkerboard, border and corner artefacts are insignificant for high-frequency images such as this one.

Figures 29c and 31c are the result of Figure 26b. This is an image of common sketches. In addition, the results show that our improvement method yielded better performance when the compression ratio was medium to low ($QP \leq 40$). PSNR increased by as much as 0.34 dB, while SSIM also slightly increased (at most 0.00982).

Figure 27 is the partial visual comparison of the logo image. Figure 27c,d is the result after linear contrast enhancement. One can clearly see that the artefacts appeared in the DPK result (c), and these have been completely eliminated by our proposed method (d).

From these experiments, we learn that under the same compression ratio, our improvement method yielded better results on low-frequency images (mostly white or monochrome, fewer variations) such as Figures 24–26. On high-frequency images (less white or monochrome parts, large variations), our performance was about the same as DPK, as seen for the image of Lena. As the checkerboard, border and corner artefacts have been eliminated, the values of neighbouring pixels became more “even”. This effectively increased the compression ratio to produce smaller files.

5. Conclusions

JPEG-XR encoding in lossy conditions causes chequerboard, border and corner artefacts. These phenomena are particularly noticeable for higher compression ratios ($QP \geq 41$). We discovered that uneven POT results are the cause of these artefacts. We therefore offer a method of improvement, which involves the use of a “uniform matrix”. This method improves the results for 4×4 , 4×1 and 2×2 POT, making them “even”. Experiments prove that this method effectively ameliorates chequerboard, border and corner artefacts while yielding the same or better image quality than the original DPK program with no increased calculation complexity or file size. Under the same compression ratio (QP), file sizes may even be smaller. Results show that the proposed improvement method is very effective.

Author Contributions: Conceptualization, K.-L.H., H.T.T., K.S., Y.-Y.C. and C.H.C.; methodology, K.-L.H., H.T.T., K.S., Y.-Y.C. and C.H.C.; software, K.-L.H., H.T.T., K.S., Y.-Y.C. and C.H.C.; validation, K.-L.H., H.T.T., K.S., Y.-Y.C., C.H.C., V.S. and A.Y.Z.; investigation, K.-L.H., H.T.T., K.S., Y.-Y.C. and C.H.C.; resources, K.-L.H., H.T.T., K.S., Y.-Y.C. and C.H.C.; writing, original draft preparation, K.-L.H., H.T.T., K.S., Y.-Y.C., C.H.C., V.S. and A.Y.Z.; review and editing, K.S., V.S. and A.Y.Z.; visualization, K.-L.H., H.T.T., K.S., Y.-Y.C., C.H.C., V.S. and A.Y.Z.; supervision, K.S., V.S. and A.Y.Z.

Funding: This work was financially supported by the Center of Intelligent Robots from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan and the Ministry of Science and Technology of Taiwan under Grants MOST107-2218-E-011-014, MOST106-2221-E-011-154-MY2 and MOST106-2218-E-011-009-MY2.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chuman, T.; Sirichotedumrong, W.; Kiya, H. Encryption-then-compression systems using greyscale-based image encryption for jpeg images. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 1515–1525. [[CrossRef](#)]
2. Schelkens, P.; Ebrahimi, T.; Gilles, A.; Gioia, P.; Oh, K.J.; Pereira, F.; Perra, C.; Pinheiro, A.M. JPEG Pleno: Providing representation interoperability for holographic applications and devices. *ETRI J.* **2019**, *41*, 93–108. [[CrossRef](#)]
3. Zhang, X.; Lin, W.; Wang, S.; Liu, J.; Ma, S.; Gao, W. Fine-grained quality assessment for compressed images. *IEEE Trans. Image Process.* **2019**, *28*, 1163–1175. [[CrossRef](#)] [[PubMed](#)]
4. Mhamdi, M.; Zribi, A.; Perrine, C.; Pousset, Y. Efficient multiple concatenated codes with turbo-like decoding for UEP wireless transmission of scalable JPEG 2000 images. *IEEE Access* **2019**, *7*, 6327–6336, doi:10.1109/ACCESS.2018.2886212. [[CrossRef](#)]
5. Yao, Y.; Hu, W.; Zhang, W.; Wu, T.; Shi, Y.Q. Distinguishing computer-generated graphics from natural images based on sensor pattern noise and deep learning. *Sensors* **2018**, *18*, 1296, doi:10.3390/s18041296. [[CrossRef](#)] [[PubMed](#)]
6. Li, R.; Duan, X.; Li, X.; He, W.; Li, Y. An energy-efficient compressive image coding for green internet of things (IoT). *Sensors* **2018**, *18*, 1231, doi:10.3390/s18041231. [[CrossRef](#)] [[PubMed](#)]
7. Ablinger, V.; Zenz, C.; Hammerle-Uhl, J.; Uhl, A. Compression standards in finger vein recognition. In Proceedings of the 2016 International Conference on Biometrics (ICB), Halmstad, Sweden, 13–16 June, 2016; pp. 1–7.
8. Liu, Y.; Xiong, Z.; Lu, L.; Hohl, D. Fast SNR and rate control for JPEG XR. In Proceedings of the 2016 10th International Conference on Signal Processing and Communication Systems (ICSPCS), Gold Coast, QLD, Australia, 19–21 December 2016; pp. 1–7, doi:10.1109/ICSPCS.2016.7843328. [[CrossRef](#)]
9. Li, L.; Zhou, Y.; Lin, W.; Wu, J.; Zhang, X.; Chen, B. No-reference quality assessment of deblocked images. *Neurocomputing* **2016**, *177*, 572–584. [[CrossRef](#)]
10. Zhan, Y.; Zhang, R. No-reference JPEG image quality assessment based on blockiness and luminance change. *IEEE Signal Process. Lett.* **2017**, *24*, 760–764, doi:10.1109/LSP.2017.2688371. [[CrossRef](#)]
11. Chen, H.; He, X.; An, C.; Nguyen, T.Q. Deep wide-activated residual network based joint blocking and color bleeding artefacts reduction for 4:2:0 JPEG-compressed images. *IEEE Signal Process. Lett.* **2019**, *26*, 79–83, doi:10.1109/LSP.2018.2880146. [[CrossRef](#)]

12. Nafchi, H.Z.; Shahkolaei, A.; Hedjam, R.; Cheriet, M. MUG: A parameterless no-reference JPEG quality evaluator robust to block size and misalignment. *IEEE Signal Process. Lett.* **2016**, *23*, 1577–1581, doi:10.1109/LSP.2016.2608865. [[CrossRef](#)]
13. Lakhani, G. Modifying JPEG binary arithmetic codec for exploiting inter/intra-block and DCT coefficient sign redundancies. *IEEE Trans. Image Process.* **2013**, *22*, 1326–1339, doi:10.1109/TIP.2012.2228492. [[CrossRef](#)] [[PubMed](#)]
14. Ozturk, E.; Mesut, A.; Carus, A. Performance comparison of JPEG, JPEG2000 JPEG XR image compression standards. In Proceedings of the 2016 24th Signal Processing and Communication Application Conference (SIU), Onguldak, Turkey, 16–19 May 2016; pp. 201–204, doi:10.1109/SIU.2016.7495712. [[CrossRef](#)]
15. Chen, H.; Guo, X.; Zhang, Y. Implementation of onboard JPEG XR compression on a low clock frequency FPGA. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 2811–2814, doi:10.1109/IGARSS.2016.7729726. [[CrossRef](#)]
16. Golestaneh, S.A.; Chandler, D.M. No-reference quality assessment of JPEG images via a quality relevance map. *IEEE Signal Process. Lett.* **2014**, *21*, 155–158, doi:10.1109/LSP.2013.2296038. [[CrossRef](#)]
17. Liu, X.; Cheung, G.; Wu, X.; Zhao, D. Random walk graph laplacian-based smoothness prior for soft decoding of JPEG images. *IEEE Trans. Image Process.* **2017**, *26*, 509–524, doi:10.1109/TIP.2016.2627807. [[CrossRef](#)] [[PubMed](#)]
18. Srinivasan, S.; Tu, C.; Regunathan, S.L.; Sullivan, G.J. HD Photo: A new image coding technology for digital photography. In *Proceedings of the SPIE Applications of Digital Image Processing XXX, San Diego, CA, USA, 26–30 August 2007*; International Society for Optics and Photonics: Bellingham, WA, USA, 2007; Volume 6696, p. 66960A.
19. Huang, W.H.; Chen, Y.Y.; Lin, P.Y.; Hsu, C.H.; Hua, K.L. Efficient and accurate document image classification algorithms for low-end copy pipelines. *EURASIP J. Image Video Process.* **2016**, *2016*, 32, doi:10.1186/s13640-016-0135-4. [[CrossRef](#)]
20. Dong, X.; Hua, K.L.; Majewicz, P.; McNutt, G.; Bouman, C.A.; Allebach, J.P.; Pollak, I. Document page classification algorithms in low-end copy pipeline. *J. Electron. Imaging* **2008**, *17*, 043011, doi:10.1117/1.3010879. [[CrossRef](#)]
21. Yu, H.; Chen, S.; Chen, Y.; Liu, H. Joint reconstruction of dynamic PET activity and kinetic parametric images using total variation constrained dictionary sparse coding. *Inverse Probl.* **2017**, *33*, 055011. [[CrossRef](#)]
22. Wang, L.; Zhang, H.; Wong, K.C.; Liu, H.; Shi, P. Physiological-model-constrained noninvasive reconstruction of volumetric myocardial transmembrane potentials. *IEEE Trans. Biomed. Eng.* **2010**, *57*, 296–315. [[CrossRef](#)] [[PubMed](#)]
23. Wang, L.; Wong, K.C.; Zhang, H.; Liu, H.; Shi, P. Noninvasive computational imaging of cardiac electrophysiology for 3-D infarct. *IEEE Trans. Biomed. Eng.* **2011**, *58*, 1033–1043. [[CrossRef](#)] [[PubMed](#)]
24. Tu, C.; Srinivasan, S.; Sullivan, G.J.; Regunathan, S.; Malvar, H.S. Low-complexity hierarchical lapped transform for lossy-to-lossless image coding in JPEG XR/HD Photo. In *Proceedings of the Applications of Digital Image Processing XXXI, San Diego, CA, USA, 10–14 August 2008*; International Society for Optics and Photonics: Bellingham, WA, USA, 2008; Volume 7073, p. 70730C.
25. Sweldens, W. The Lifting Scheme: A new philosophy in biorthogonal wavelet constructions. In Proceedings of the SPIE Wavelet Applications in Signal and Image Processing III, San Diego, CA, USA, 9–14 July 1995; International Society for Optics and Photonics: Bellingham, WA, USA, 1995; Volume 2569, pp. 68–79.
26. Corp, M. HD Photo Device Porting Kit 1.0. Available online: [https://docs.microsoft.com/en-us/previous-versions/windows/hardware/download/dn550976\(v=vs.85\)](https://docs.microsoft.com/en-us/previous-versions/windows/hardware/download/dn550976(v=vs.85)) (accessed on 1 January 2019).
27. Sweldens, W. The lifting scheme: A construction of second generation wavelets. *SIAM Jo. Math. Anal.* **1997**, *29*, 511–546. [[CrossRef](#)]
28. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
29. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. The SSIM Index for Image Quality Assessment. Available online: <https://ece.uwaterloo.ca/~z70wang/research/ssim/> (accessed on 1 January 2019).
30. The Academy of Motion Picture Arts and Sciences. The 84th Academy Awards Poster. Available online: <https://www.imdb.com/title/tt2089826/mediaviewer/rm94416896> (accessed on 1 January 2019).

31. Gallaway, J. James Joyce (Sketch From a Famous Photo of James Joyce). Available online: <http://www.flickr.com/photos/63923085@N00/2304643567> (accessed on 1 January 2019).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).