Suppressing artifacts in block DCT coded images based on re-encoding, regression, and image prior

Younghee Kwon¹       Kwang In Kim²
KAIST             Saarland University
Jin Hyung Kim³
KAIST
CS/TR-2008-293
November, 2008

KAIST
Department of Computer Science

This article can be found in the KAIST CS/TR Archives for 2007-2008,

¹kyhee@ai.kaist.ac.kr
²kimki@tuebingen.mpg.de
³jkim@cs.kaist.ac.kr
Suppressing Artifacts in Block DCT Coded Images
Based on Re-encoding, Regression, and Image Prior

Younhee Kwon, Kwang In Kim, and Jin H. Kim

Abstract. Post-processing a block-based discrete cosine transform (BDCT) encoded image requires solving two seemingly contradictory tasks of suppressing discontinuities at block boundaries and enhancing edge and texture details. This paper approaches this problem by combining the existing algorithms which are specialized to each individual task. The re-application of JPEG applies BDCT coding to pixel-wise shifted versions of the input BDCT encoded image and takes the average of these re-encoded images after shifting them back to the original positions. This step effectively restores continuities in block boundaries while tending to keep the existing details. Then, a set of regressors are trained based on example pairs of these artifact-suppressed images and the corresponding clean images such that missing high-frequency details lost during the DCT are restored. Furthermore, two generic image models which take into account the leptokurtic nature of natural images in the wavelet and spatial domains, respectively are adopted so that block artifacts remaining after the regression step are removed and the major edges are enhanced. Comparison with the existing post-processing methods shows the effectiveness of the proposed method.

1 Introduction

Block-based discrete cosine transform (BDCT) coding is one of the most widely used tools for encoding still images (e.g., JPEG) and video sequences (e.g., MPEG). In BDCT coding, an image is partitioned into small blocks (typically of size $8 \times 8$) which are independently transformed using the DCT, quantized, and encoded. The BDCT has appealing properties such as energy compaction, orthogonality that leads to decorrelation, and efficiency in computation. However, at low bit rates, the BDCT encoded images can exhibit discontinuities appearing between the boundaries of the blocks, known as block artifacts. This problem can be avoided by suitably preprocessing images (e.g., diffusion preprocessing [1] and learning-based methods [2]) before the encoding process or by using other coding standards which do not show block artifacts (e.g., JPEG2000). However, these methods might suffer from different kinds of encoding artifacts and more importantly, require changing the existing standards. In this respect, post-processing appears to be a more practical approach: it does not require modifying encoding methods and can be applied directly to encoded images which are not available in the unencoded form. In this paper, we propose a method of post-processing BDCT encoded images with an emphasis on enhancing the JPEG encoded images at low bit rates. The remainder of this section reviews existing algorithms and briefly describes the main idea of the proposed method.

The suppression of block artifacts is not only practically important but also theoretically interesting as it provides an important example of denoising images contaminated by non-Gaussian noise. Accordingly, various methods have proposed from different backgrounds. One of the first methods is based on image filtering. Reeve III and Lim [3] applied low-pass filtering (or blurring) to the pixels at the block boundary. This method can efficiently suppress block artifacts. However, at the same time, it introduces unnecessary blurring of edge and texture details. Adaptive filtering tends to avoid this problem by locally adjusting the degree and direction of the blur. Tschumperlé and Deriche demonstrated that generic anisotropic diffusion filtering [4] is already effective in block artifact suppression. In a more JPEG-specific direction, based on an estimate of the quantization error, Chou et al. [5] classified image edges into artificial edges (block artifacts) and natural edges, which can guide the adaptation of blur process. This idea was also adopted in the context of soft-thresholding-based image denoising in the wavelet domain: Xiong et al. [6] exploited cross-scale correlation of wavelet coefficients to detect natural edges which are then used to determine a proper threshold in each wavelet band. In the same domain, Liew and Yan [7] analyzed statistical characteristic of block boundaries which resulted in more accurate estimation of thresholds.
As in other fields of image enhancement, utilizing \textit{a priori} knowledge of image class is essential in block artifact suppression. The theory of projection onto convex sets (POCS) is a widely used tool in this context. The underlying idea is to represent \textit{a priori} knowledge of natural images as a sequence of convex constraints (more specifically, a sequence of convex sets whose elements satisfy certain constraints corresponding to known characteristic of natural images). Then, for a given image, the element which is closest to that image and satisfies all the constraints can be retrieved by simple iteration through the projection onto each individual set. Commonly used constraints include spatial smoothness \cite{8}, quantization constraints, etc. A rather direct way of utilizing \textit{a priori} knowledge is to encode it into a distribution or an energy functional. Sun and Cham \cite{9} proposed a \textit{maximum a posteriori} (MAP) framework where the prior is modeled as a Markov random filed (MRF). Then, the clique potential for MRF was learned from a set of natural images based on the fields of experts, which led to an improved performance over several existing methods including those methods based on POCS and overcomplete wavelet representation.

Nosratinia \cite{10} proposed another promising method called \textit{re-application of JPEG}. This algorithm generates a set of pixel-wise shifted versions of the input JPEG image, re-applies JPEG encoding to the shifted versions, and shifts them back to the original positions. Then, the denoised image is obtained by simply averaging these re-encoded images. Despite its simplicity, \textit{re-application of JPEG} demonstrated superior performance over the algorithms based on nonlinear filtering, POCS, and overcomplete wavelets.

The success of machine learning in many computer vision and image enhancement applications also inspired a new class of artifact suppression algorithms. The basic idea is to estimate a function which maps the given JPEG encoded image to the desired clean image, based on example pairs of JPEG encoded images and the corresponding uncompressed images. Qiu \cite{11} used a multilayer Perceptron which receives the gradient of pixel values in an one-dimensional cross section across the block boundary and produces an estimation of difference between the encoded image and the original image. A similar algorithm was also developed in the MAP framework by Yang et al. \cite{12}. Lee et al. \cite{13} proposed performing the piecewise linear regression in the space of DCT coefficients and showed comparable results to those of \textit{re-application of JPEG}. Recently, Kim and Kwon \cite{14} proposed casting the artifact removal to the image super-resolution problem by blurring the input JPEG image. The super-resolution is then performed by a patch-wise regression between blurred images and missing high-frequency details. This method demonstrated an improved performance over several existing methods including the piecewise linear regression in DCT domain, \textit{re-application of JPEG}, and the adaptive total variation minimization-based method (cf. references appearing in \cite{14}). However, there is still much room for improvement: as an intermediate step, this method firstly blurs the input image, which actually generates a degraded image (i.e., worse than the input JPEG image) on which the subsequent processing steps are based. Furthermore, as an instance of a generic framework, it does not fully utilize domain knowledge, which could boost the performance (e.g., the block structure of DCT coding; cf. Sec. 2.2).

Post-processing a JPEG encoded image requires solving two seemingly contradictory tasks of suppressing block artifacts and enhancing the existing details. In principle, one could build a single large image model which simultaneously solves these tasks. However, more economical approach might be to resolve each sub-problem using an algorithm specially tailored for it. The basic idea of the proposed method is to combine the \textit{re-application of JPEG} \cite{10} and the super-resolution-based method \cite{14}. The \textit{re-application of JPEG} (henceforth referred to as \textit{`re-encoding'}) effectively suppresses the block boundary while tending to keep the existing details. Then, a set of regressors trained on an example set of artifact-suppressed images and the corresponding clean images, recovers high-frequency details which are lost during the DCT. These two steps can already produce significantly enhanced images. However, none of them completely resolve each sub-problem: some block artifacts are still remaining in the results of the re-encoding (cf. the second row of Fig 5) and for steep major edges, the regressors tend to leave them blurred or generate ringing artifacts to compensate the loss of smoothness (cf. Fig 1.c). Generic image models which take into account the prior over the structure of natural images are then adopted to post-process the results.

This paper is organized as follows. Section 2 presents the components of the artifact suppression system in the order of system flow. Experimental results are presented in section 3 while conclusions are given in section 4.

2 Suppression of Block Artifacts and Enhancement of Details

The proposed method is composed of four steps. Fig. 1 illustrates these steps with an example. In the first step, the input JPEG image is pre-processed based on re-encoding. While this step effectively suppresses the block boundaries, it does not enhance any existing details, which is then achieved by the regression-based refinement (the second step). The third step exploits \textit{a priori} knowledge of natural images to post-process the result of regression.
so that remaining block artifacts are further suppressed while the steep edges are enhanced. The last optional step sharpens the resulting images by introducing artificial high-frequency details.

2.1 Re-encoding

Within each quantization block, DCT encoding has properties of low-pass filtering plus making the result a stationary point of the encoding, i.e., repeated encoding does not alter the encoded block. The underlying idea of the re-encoding (re-application of JPEG) is that, when applied to shifted images, BDCT has an effect of suppressing the block boundary which in the shifted version, contributes to the high-frequency components. At the same time, it tends to keep the existing details inherent within the original block since the high-frequency components were already removed from them and accordingly they must be already close to a stationary point of DCT encoding.

As a pre-processor for regression, an important feature of the re-encoding is that the entire data processing is spatially localized. In general, one might not apply regression directly to the whole image unless the class of images of interest is significantly restricted (e.g., focused faces and digits). Besides the obvious fact that the space of input data are not vectorial (i.e., they are of different sizes), this setting has the drawback of making the dimensionality of regression space too large, which can lead the regressors into overfitting. Accordingly, we adopt a local patch-based method where the input image is decomposed into small sub-windows (patches) and each patch is processed independently of other patches [15, 14]. If either one of encoding or pre-processing steps were not spatially localized, applying a local-patch based method might not be so straightforward since in this case, image locations far away from the given patch of interest can affect the generation of that patch and by construction, the regression part cannot take this global context into account.¹

2.2 Regression-based Refinement

This step builds upon the local patch-based method of Kim and Kwon [14] where for each pixel location in the input image, a regressor receives a patch (of size $\sqrt{M} \times \sqrt{M}$) centered at that location and produces an estimate of the corresponding patch (of size $\sqrt{N} \times \sqrt{N}$) in the desired image. While a single regressor was trained for the

¹Certainly, this does not imply that taking into account the global context in JPEG artifact removal is meaningless. Actually, we will exploit this in the later stage (cf. Sec. 2.3).
observed in the later case in the preliminary experiments. From the images used in training and testing, \( KRRs \) are then found by minimizing the cost functional (1) using the combination of the kernel matching pursuit and estimates and the truncation at the boundary of image range (i.e., errors corresponding to out of range \((0, 255)\) values can be corrected by truncation and accordingly do not actually contribute to errors in the image domain).

Training image pairs are obtained by JPEG encoding a set of clean images followed by the re-encoding. Then, the training examples for the \((y, x)\)-th regressor \((1 \leq y, x \leq 8)\) are sampled from the corresponding index sets \({\{(a \ast 8 + y, b \ast 8 + x)\}}\) of pixel locations where \(a\) and \(b\) are random nonnegative integers so that the index set is bounded in the size of each example image. Similarly to \([14, 15]\), the input image is firstly band-frequency filtered based on the Laplacian of Gaussian (LOG) filter. Then, given a patch of LOG filtered image, the regressor estimates a patch corresponding to the difference between the result of re-encoding and the underlying ground truth that the final output is obtained by adding the regression result to the re-encoded image. When \(N > 1\), the output patch is overlapping to its spatial neighbors, which constitutes a set of candidates for each pixel location. The image-valued output is constructed by taking the average of these candidates for each pixel.\(^2\)

As a pre-processing, the dimensionality of input data is reduced based on principal component analysis (PCA). Then, for a given set of pre-processed data points \({\{(x_1, y_1), \ldots, (x_l, y_l)\}} \subset \mathbb{R}^P \times \mathbb{R}^N\) \((P\) is the dimensionality of PCA subspace), we minimize the following regularized cost functional for the regressor (referred to as \(kernel\ ridge\ regression:\) \(KRR\)) \(f = \{f^1, \ldots, f^N\}:^3\)

\[
O(f) = \sum_{i=1,\ldots,N} \|f(x_i) - y_i\|^2 + \lambda \sum_{j=1,\ldots,l} \|f^j\|^2_{\mathcal{H}},
\]

where the first norm is the usual \(L^2\)-norm and \(\mathcal{H}\) is the \(reproducing\ kernel\ Hilbert\ space\) (RKHS) generated by a Gaussian kernel

\[
k_{\sigma_k}(x, y) = \exp\left(-\frac{\|x - y\|^2}{\sigma_k}\right).
\]

If there’s no restriction on the space of solutions, the optimum \(\{f^j\}\) is expanded in training data points (i.e., \(f^i() = \sum_{j=1,\ldots,l} a_j^i k(x_j, \cdot), \) for \(i = 1, \ldots, N\)). However, to reduce the computational complexities of training and testing for the case of using a large database \((l \approx 200,000\) in our experiments), we restrict the possible solutions in the following form

\[
f^i() = \sum_{j=1,\ldots,l_b} a_j^i k_{\sigma_k}(b_j, \cdot), \) for \(i = 1, \ldots, N,
\]

where \(l_b\) is fixed at 300 throughout the current paper. The basis points \(\{b_j\}\), the kernel function \(k_{\sigma_k}\), and the regularization parameter \(\lambda\) are shared by \(N\) regressors such that the computational complexities of training and testing all the regressors remain roughly the same to those of a single regressor. The parameters \(\{a_j^i\}\) and \(\{b_j\}\) are then found by minimizing the cost functional (1) using the combination of the kernel matching pursuit and gradient descent (GD) as proposed in \([14]\). However, instead of parameterizing both \(\{a_j^i\}\) and \(\{b_j\}\) for the GD (as proposed in \([14]\)), we perform the GD only in \(\{b_j\}\) and in each GD step, calculate \(\{a_j^i\}\) analytically for a fixed value of \(\{b_j\}\). While both methods are guaranteed to converge to local optima,\(^4\) a faster speed of convergence was observed in the later case in the preliminary experiments.

The parameters including the input patch size \((M)\), the output patch size \((N)\), and the (hyper-)parameters for \(KRRs\) \((\sigma_k, \) and \(\lambda\) for each location \((y, x)\)) are optimized based on a set of validation images which are disjoint from the images used in training and testing.\(^5\) while the dimensionality of PCA subspace \(P\) is simply determined

---

\(^2\)As demonstrated in \([14]\), a better alternative is to take a general convex combination (i.e, non-uniform averaging) where the combination coefficients are estimated from the local context of neighborhood candidates. However, in this setting, the optimization of hyper-parameters for the array of regressors becomes computationally very dense.

\(^3\)For the simplicity of notation, we omitted the spatial index \((y, x)\) from the regressor symbol \(f\). It should be noted that since the output is a patch, for each location \((y, x)\), \(N\) different regressors (or equivalently, an \(N\)-dimensional vector-valued regressor) are trained.

\(^4\)The less obvious proof of the convergence for the later case is provided in \([16]\).

\(^5\)There are several standard methods for optimizing the parameters for \(KRR\), including generalized cross-validation and marginal likelihood maximization \([17, 18]\). However, these method should not be directly applied for optimizing the individual \(KRRs\) used in the current setting as they do not take into account the further regularization effects of averaging overlapping estimates and the truncation at the boundary of image range (i.e., errors corresponding to out of range \((0, 255)\) values can be corrected by truncation and accordingly do not actually contribute to errors in the image domain).

---

4
Table 1: Quantization tables used in JPEG encoding

<table>
<thead>
<tr>
<th></th>
<th>Q1</th>
<th>Q2</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>60 70 70 90 120 255 255</td>
<td>110 130 150 192 255 255 255</td>
</tr>
<tr>
<td>60</td>
<td>60 70 96 120 255 255 255</td>
<td>130 150 192 255 255 255 255</td>
</tr>
<tr>
<td>70</td>
<td>70 80 120 200 255 255 255</td>
<td>150 192 255 255 255 255 255</td>
</tr>
<tr>
<td>70</td>
<td>96 120 145 255 255 255 255</td>
<td>192 255 255 255 255 255 255</td>
</tr>
<tr>
<td>90</td>
<td>130 200 255 255 255 255 255</td>
<td>255 255 255 255 255 255 255 255</td>
</tr>
<tr>
<td>120</td>
<td>255 255 255 255 255 255 255</td>
<td>255 255 255 255 255 255 255 255</td>
</tr>
</tbody>
</table>

at the smallest integer containing 99% of total variances in the whole space. To facilitate the optimization of KRR specific parameters, we firstly estimate the optimal parameters for a single regressor which is independent of the block structure (i.e., its training examples are sampled randomly from the entire image region). The parameters for the array of regressors are firstly initialized with those parameters. Then, the $(8 \times 8)$-sized block is scanned in raster order such that at the location $(y, x)$, the parameters for the $(y, x)$-th regressor is optimized for given fixed other regressors. We have not obtained any theoretical guarantee of the convergence of this setting. However, we have empirically observed that around 10 sweeps through the entire set of regressors are already sufficient for producing the convergence behavior.

Similarly to [14], the application scenario of the proposed method is to train a model specialized to each small interval of compression factors such that the whole range of compression factors is covered by several models. Then, removing artifacts of a given input JPEG image can be performed by choosing the proper model based on its compression factor which is specified by the quantization table stored at the header of JPEG file. However, to facilitate the comparison with existing methods, we focus on only two specific quantization tables for JPEG encoding (Q1 and Q2 in Table 1) such that both in training and in testing, the JPEG images are obtained with these quantization tables (one model for each table). Application to other compression factors should be straightforward. In this setting, $M$, $N$, and $P$ were determined at 7, 5, and 20, respectively for Q1 and at 9, 5, and 55, respectively for Q2. For the case of Q2, $\{\sigma_k^{(y,x)}\}$ and $\{\lambda_k^{(y,x)}\}$ ranged in $[0.24 \cdot 10^{-3}, 0.54]$ and $[0.1 \cdot 10^{-6}, 0.78 \cdot 10^{-2}]$, respectively.

2.3 Exploiting Prior on Natural Images

As shown in Fig. 1.c, the regression stage already produces significantly improved images over the original JPEG encoded images and the results of re-encoding. However, it should be noted that the regression stage is regularized only by a generic prior corresponding to a Gaussian kernel and accordingly does not take into account known specific characteristic of natural images, which might have guided better the reconstruction of images. For instance, the regression stage fails to recover steep major edges: it leaves them blurred or produces ringing artifacts (cf. the enlarged sub-images of Fig. 1.c) since the abrupt changes of function values contribute to high-order derivatives and are penalized severely by the regularizer. Furthermore, since the parameters for the regressors are not specifically tailored to each local image region but are tuned to minimize the overall $L^2$ error, some block artifacts (remaining after the re-encoding step) can still remain. These problems could be remedied by locally adjusting the parameters. However, more direct approach is to exploit the prior knowledge of which images are more likely to be natural. While there are numerous results on the statistical properties of natural images and methods of utilizing them for image enhancement, we focus on two specific image priors which have shown promising results in related applications and turned out to be effective in the above-mentioned problems.

The product of edge-perts (PoEdges) [19] characterizes coefficients (the representation of images) in the wavelet domain with leptokurtic distributions and models higher order dependencies between them through the product of experts model. This model effectively captures the clustering behavior of wavelet activities [20] as the resulting sparsity prior over the joint distribution prefers simultaneous activation of few coefficients in nearby scales and spatial locations. It has shown promising results in a specific domain of image denoising with emphasis on Gaussian noise [19]. However, since it is a generic image prior, in principle, it can be used in any image enhancement applications. For the Gaussian noise case, PoEdges model provides a maximum a posteriori (MAP) framework in
the decorrelated wavelet domain:

$$\tilde{z}^* = \arg \max_{\tilde{z}} \left( \log p(\tilde{x}|\tilde{z}) + \log p(\tilde{z}) \right)$$

$$= \arg \min_{\tilde{z}} \left( \frac{1}{2} \| \tilde{z} - \tilde{x} \|^2 + \sigma_p \left( \sum_j w_j \tilde{z}_j^2 \right)^{\alpha_p} \right),$$

where $$\tilde{x} = \mathcal{W}x$$, $$\mathcal{W}$$ is the wavelet transform, $$x$$ is the input noisy image, the parameterization of experts model $$\{w_j\}$$ is estimated by an expectation maximization type algorithm [19], and the hyper-parameter $$\alpha_p$$ controlling the decaying shape of the responses of experts is fixed at 0.5. A straightforward approach for the application of PoEdges to JPEG artifact suppression is to modify the noise model according to the JPEG encoding process:

$$\tilde{z}^* = \arg \min_{\tilde{z}} \left( \frac{1}{2} \| \mathcal{W}(\mathcal{J}(\mathcal{W}^\#(\tilde{z}))) - \tilde{x} \|^2 + \sigma_p' \left( \sum_j w_j \tilde{z}_j^2 \right)^{\alpha_p} \right),$$

where $$\mathcal{W}^\#$$ is the (pseudo) inverse of $$\mathcal{W}$$ and $$\mathcal{J}$$ denotes the JPEG encoding operator. However, $$\mathcal{J}$$ is non-differentiable and is not even continuous which can cause difficulty in optimization.\(^6\) Accordingly, for the computational efficiency, we simply penalize the deviation from the results of regression ($$y$$):

$$\tilde{z}^* = \arg \min_{\tilde{z}} \left( \frac{1}{2} \| \tilde{z} - \mathcal{W}y \|^2 + \lambda_p \left( \sum_j w_j \tilde{z}_j^2 \right)^{\alpha_p} \right). \quad (2)$$

Unfortunately, in this case, intuitive probabilistic interpretation of denoising process is not any more possible. Instead, one could regard (2) as a regularization framework where the regularizer enforces sparsity in the wavelet domain.

The PoEdges turned out to be especially helpful in removing the remaining block boundaries after regression. In part, this can be explained by its smoothing behavior in the wavelet domain: choosing sparsely the wavelet bases results in suppressing a certain type of signal, which up to a certain degree must have low-pass filtering effect. Furthermore, due to the clustering behavior of wavelet activities, PoEdges might tend to suppress signals that are not occurring across the neighboring scales and orientations, which are more likely to be block boundaries than natural edges. However, due to the very same property, it is unlikely to stress very high-frequency signals, and accordingly is inappropriate for modeling steep major edges. To process steep major edges properly, a model which can deal with extra-high frequency signals is required. The leptokurtic property of derivative distribution in the spatial domain leads to the natural image prior (NIP) [22]:

$$z^* = \arg \min_z \left( \| z - y \|^2 + \lambda_N \sum_{(j,i) \in N_s(j)} \left( |z_j - z_i| \right)^{\alpha_N} \right), \quad (3)$$

where $$N_s(j)$$ is the 8-connected neighbors of the pixel location $$j$$. Here, the first summand penalizes the deviation from the regression result while the second summand (referred to as natural image prior) for $$\alpha_N < 1$$ enforces sparsity in the gradient domain such that a narrow sharp edge is preferred over a spread fuzzy edge.\(^7\) Accordingly, this model can well-represent the abrupt change of pixel values. However, since it tends to flatten textured area, we only apply NIP at major edges which are found by thresholding each pixel based on the $$L^2$$ norm of the Laplacian and the range of pixel values in the local patches (i.e., classifying a pixel into ‘major edge class’ if the norm of Laplacian and the maximum difference of pixel values within a local patch are larger than thresholds $$T_{M1}$$ and $$T_{M2}$$, respectively). The optimization of (3) is performed by the belief propagation (BP). To facilitate the optimization, we use the candidates generated during the regression step (i.e., those candidates which are averaged to construct the image output) such that the best candidates are chosen for each pixel by the BP.

\(^6\)Another possibility leading to a probabilistic framework is to adopt a differentiable estimation of encoding noise as proposed by Robertson and Stevenson [21].

\(^7\)The original NIP model was proposed for image super-resolution and resulted in a MAP algorithm by adopting a noise model which takes into account the generation of low-resolution images [22]. Here, we use a simplified variation presented in [14].
The parameter $\alpha_N$ in the NIP was fixed at 0.85 throughout the current paper while the other parameters were determined based on validation images in terms of PSNR values: $\lambda_P$ for PoEdges and $\lambda_N$ for NIP were determined at 5 and 2.55, respectively for Q1 and 4 and 5.1, respectively for Q2 while the thresholds values $T_{M1}$ and $T_{M2}$ used in finding the major edges were calculated as 689 and 445, respectively for Q1 and 383 and 255, respectively for Q2.

2.4 Example-based Sharpening

The last step of the proposed method is to sharpen the results of the previous step. This is motivated by our preliminary experiments on the subjective visual evaluation of images, which indicated that for low-quality JPEG encoded images, artificially increasing the contrast (of the results of the previous step) can generate visually more plausible images even though it leads to lower PSNR values.

While the image sharpening can be performed by simple high-boost filtering, we adopt the example-based super-resolution method proposed by Freeman et al. [15] which provides more coherent enhancement of edges. The underlying idea is to replace each small patch of the input blurry image by a stored high-contrast patch whose blurred version is closest to the input patch. The distance measure is designed so that the compatibility of a reconstructed patch with its spatial neighbors are retained (readers are referred to [15] for details). Actually, this method is similar to our regression step. The main difference is that in this nearest neighbor (NN)-based scheme, the output is one of the example patches. Accordingly, at least locally, the range of processing is confined in the space of natural images while this is not the case for the KRR which synthesizes the output. To facilitate the introduction of high-contrast components in the resulting image, example patches are only sampled at the edges. Furthermore, during the training of this stage, the input images are slightly blurred such that when applied directly to the results of the previous step, it over-stress the high-frequency components. As observed in Figs. 1, 3-5, the sharpening significantly enhances the edges and accordingly can offer improved visual plausibility. However, in general, a preference between sharpened and unsharpened images might be subjective and depend on the specific image. Furthermore, for the case of experiments with Q2, the sharpening resulted in on average 0.9 decrease of PSNR values from the results of regression, which made the results even worse than the input JPEG images in PSNR. Accordingly, this step should be regarded as an optional component.

3 Experiments

To evaluate the performance of the proposed method, we used twelve 512 × 512-sized images, most of which are commonly used in the evaluation of computer vision and image processing algorithms. The images are encoded using the cjpeg Linux software based on the quantization tables listed in Table 1. These tables are adopted by many published artifact suppression works (e.g., [6, 8, 7, 13, 23, 9]) and accordingly for the standard images (e.g., Lena and Goldhill in Fig. 2 (the first and second images) and Barbara in Fig. 1), the performance of the proposed method can be compared with these methods. For a rather direct comparison, Nosratinia’s re-application of JPEG [10] and Kim and Kwon’s super-resolution-based method [14] were also tested. It should be noted that both methods already outperformed many existing algorithms.

---

8The input image is scaled into the interval of [0, 255].

9However, in our preliminary experiments, replacing the proposed regression step with the NN-based method resulted in much more noisy reconstructions. A similar observation has been reported in [14].
Figures 3-5 show examples of artifact suppression. Re-application of JPEG significantly reduced block artifacts which improved both the visual quality and PSNR values. However, averaging differently encoded blocks resulted in slightly blurred edges and texture details. Super-resolution-based method produced much shaper edges and texture details. However, as observed in the lower contour of banana (Fig. 5) and in the boundary of cheek of Lena (Fig. 3) in Q2 case, it failed to remove some of block artifacts. This can be attributed to the use of the homogeneous blurring followed by a single fixed regressor both of which are tuned to minimize the overall error rates and are not sufficient to completely remove different degree of blockiness while keeping the other details. As suggested in [14] this could be resolved by using multiple blur kernels and corresponding regressors. However, it is not so straightforward how to optimize the blur kernels such that subsequent steps can be facilitated. On the other hand, the results of the proposed method show sharp and coherent texture and edge details as the results of the super-resolution-based method. At the same time, the block boundaries observed in the results of [14] are successfully removed. For quantitative evaluation, improvements of PSNR values\(^{10}\) of the results of different algorithms from the input JPEG images are plotted in Fig. 6.

4 Conclusion

In this paper, we proposed a method of post-processing BDCT encoded images. The proposed method is constructed as a step-wise combination of several existing algorithms which are developed for the specific application of JPEG artifact suppression or as generic tools for image enhancement. The re-encoding effectively suppresses the block boundary. Then, the missing high-frequency details are recovered by regression-based refinement. Two aspects of prior knowledge on natural images are adopted so that block artifacts remaining after the regression step are suppressed and the major edges are enhanced. These steps already demonstrated a comparable performance over the existing state-of-the-art methods. The visual plausibility of the resulting image was further enhanced by adopting example-based image sharpening.

Similarly to most of other existing methods, we approached the post-processing of JPEG images in the error minimization perspective. This resulted in effective removal of block artifacts and enhancement of the existing details. However, except for the optional last stage,\(^{11}\) the proposed method did not introduce any new texture details in the results, which might have produced more photo-realistic images. Since the problem of recovering texture details is severely ill-posed, it might not be well-resolved by using only the error minimization framework. On the other hand, usually the ultimate goal of post-processing is constructing visually plausible images rather than minimizing a certain error measure. Synthesizing and transferring pseudo texture details might be a good candidate in this view. The future work should focus on combination of error minimization methods including the proposed method with texture synthesis methods.

Acknowledgment

The ideas presented in this study have greatly profited from discussions with Peter V. Gehler and Daewon Lee.

References


\(^{10}\) Assuming that the range of pixel values in the entire image space are bounded in \([0, 255]\), we use the below given definition of PSNR for an observation vector \(\mathbf{N} = [N^1, \ldots, N^K]\) over the ground truth vector \(\mathbf{O} = [O^1, \ldots, O^K]\):

\[
\text{PSNR}(\mathbf{O}, \mathbf{N}) = 10 \log \left( \frac{255^2}{\sqrt{\sum_{i=1}^{K} (O^i - N^i)^2 / K}} \right). \tag{4}
\]

This should not be confused with another commonly used definition of PSNR where the range of pixel values are measured in \(\mathbf{O}\):

\[
\text{PSNR}'(\mathbf{O}, \mathbf{N}) = 10 \log \left( \frac{\max\{O^i\} - \min\{O^i\}}{\sqrt{\sum_{i=1}^{K} (O^i - N^i)^2 / K}} \right). \tag{5}
\]

\(^{11}\) Actually, the last stage not only improved the contrast but also introduced certain new texture details. However, the image area showing this effect is limited and depends heavily on the specific image, which is not easily predictable.
Figure 3: Example of artifact suppression for Lena image, a. input JPEG images, b. re-application of JPEG [10], c. super-resolution [14], d. the proposed method, and e. sharpened results of d. Increases of PSNRs from the input JPEG images (displayed below each image) were calculated based on the complete images. Please refer to the electronic version of the current paper for better visualization.
Figure 4: Example of artifact suppression for Goldhill image. a. input JPEG images, b. re-application of JPEG [10], c. super-resolution [14], d. the proposed method, and e. sharpened results of d. Increases of PSNRs from the input JPEG images were calculated based on the complete images. Please refer to the electronic version of the current paper for better visualization.
Figure 5: Example of artifact suppression, from top to bottom, input JPEG images, re-application of JPEG [10], super-resolution [14], the proposed method, and sharpened results of the output of the proposed method. Please refer to the electronic version of the current paper for better visualization.
Figure 6: Performance of different JPEG artifact removal algorithms.


