EUROGRAPHICS Workshop on Multimedia (2004)
N. Correia, J. Jorge, T. Chambel, Z. Pan (Editors)

A Training-Based Method for Reducing Ringing Artifact in BDCT-Encoded Images

Guangyu Wang,† Tien-Tsin Wong,‡ and Pheng-Ann Heng§

Department of Computer Science & Engineering
The Chinese University of Hong Kong

Abstract

The quantization procedure of block-based discrete cosine transform (BDCT) compression (such as JPEG) introduces annoying visual artifact. In this paper, we propose a novel training-based method to reduce the ringing artifact in BDCT-encoded high-contrast images (images with large smooth color areas and strong edges/outlines). Our main focus is on the removal of ringing artifact that is seldom addressed by existing methods. In the proposed method, the contaminated image is modeled as a Markov random field (MRF). We 'learn' the behavior of contamination by extracting massive number of artifact patterns from a training set. To organize the extracted artifact patterns, we use the tree-structured vector quantization (TSVQ). Instead of post-filtering the input contaminated image, we synthesize *an artifact-reduced image. We show that substantial improvement (both statistical and visual) is achieved using the proposed method. Moreover, since our method is non-iterative, it can remove artifact within a very short period of time.*

1. Introduction

The block-based discrete cosine transform (BDCT) is the core of many current image and video compression standards, such as JPEG [PM93] and MPEG [RH96]. In particular, JPEG has been a popular image standard since early 90's. Even though we now have wavelet-based JPEG2000, there are still many existing images already encoded using JPEG. Unfortunately, BDCT-based JPEG has a drawback, which is the annoying visual artifact. Such artifact is especially apparent in *high-contrast images* compressed at low bit rates. We refer high-contrast images as the ones with large smooth color regions and strong edges/outlines. A typical example is cartoon image (Fig. 1(a)).

Since 1992, JPEG has been established as the international standard for still image compression. An image is first subdivided into blocks of 8×8 pixels and each block is transformed from spatial domain to frequency domain using discrete cosine transform (DCT). The DCT coefficients are then quantized and run-length encoded. In principle, DCT does not introduce artifact. It merely transforms the original im-

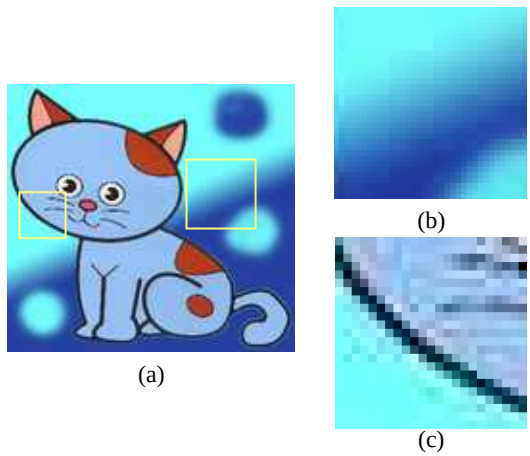


Figure 1: *BDCT compression artifact. (a) A typical BDCT-encoded (contaminated) high-contrast image; (b) the blocking artifact; (c) the ringing artifact.*

† gywang@cse.cuhk.edu.hk
‡ ttwong@acm.org
§ pheng@cse.cuhk.edu.hk

age to a domain in which it can be more effectively encoded. However, the preceding quantization of DCT coefficients is lossy and may introduces visual artifact.

There are mainly two types of artifact, namely the *blocking artifact* and the *ringing artifact*. Excessive quantization of low-frequency coefficients introduces the blocking artifact, which exhibits color discontinuity at the block boundary, as shown in Fig. 1(b). Excessive quantization of high-frequency coefficients causes the ringing artifact around strong edges [WLY02], such as outlines of cartoon character (Fig. 1(c)). Ringing artifact may spread within the 8×8 block. Most existing methods are tailored for removing blocking artifact only. In this paper, we propose a novel method to reduce ringing artifact.

Both subjective and objective qualities of BDCT-encoded images can be significantly improved by reducing BDCT compression artifact. In general, existing techniques for removing BDCT compression artifact can be roughly divided into three categories: post-filtering, projection onto convex sets (POCS), and training-based methods.

As the visual artifact in a contaminated image is usually high-frequency artifact, a straight-forward solution for reducing such artifact is to apply low-pass filtering. Reeve and Lim [RL84] first proposed a space-invariant low-pass filtering method. However such filtering often causes the loss of high-frequency details, such as edges, in the original image. Therefore, a number of adaptive spatial filtering techniques [CWQ01, KH95, LKP98, MZ95] have been proposed to overcome this problem. Projection onto convex sets (POCS) [Com93] is a theory widely used for artifact removal [WLY02, Zak92, JKK00, PKL98, YG97, LCPH96]. This method updates the BDCT coefficients by iteratively projecting onto several constraints, which are formulated by a *priori* knowledge of contaminated images. In recent years, training-based methods [CHS98, Qiu00] have been proposed. These methods only process pixels at the block boundary and do not handle ringing artifact within the block.

In this paper, we aim at removing ringing artifact in BDCT-encoded high-contrast images. We propose a training-based method which *synthesizes* the image with less artifact, rather than post-filters the BDCT-encoded image. A training set is first prepared to provide *priori* knowledge that assists us to reduce BDCT compression artifact. The training set consists of pairs of original images (images before compression and without any artifact) and BDCT-encoded images (contaminated images). We model the artifact as a Markovian random field. To do so, we extract a set of *artifact patterns* from the training set. These massive artifact patterns are looked up during the synthesis. To manage and query such mass of patterns efficiently, we organize them using the tree-structured vector quantization (TSVQ). As the ringing artifact usually appears near the strong edges (Fig. 1), we divide the input contaminated image into edge and non-edge regions. For the edge region, we synthesize each pixel value

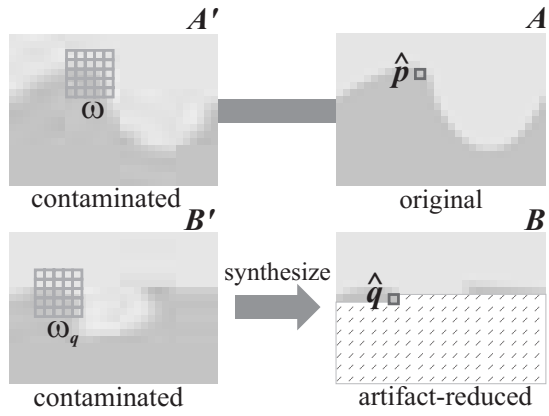


Figure 2: *The basic idea.*

by querying the TSVQ with its neighborhood as a search key. For the non-edge region where the ringing artifact is not apparent, we simply copy the pixel values from the input image.

The rest of this paper is organized as follows. Section 2 explains the rationale and basic idea of our method. Section 3 describes the details of our proposed training-based method. Section 4 evaluates the proposed method. Finally, conclusions are drawn in Section 5.

2. Contamination as Markov Random Field

To reduce the visual artifact due to BDCT-encoding, we need to understand its source. During BDCT-encoding, a 8×8 block of pixel values, $f(u,v)$, are transformed to frequency domain by the discrete cosine transform. Each DCT coefficient is then quantized according to a fine-tuned quantization table in JPEG standard. DCT itself does not introduce error, but the preceding quantization does.

During DCT, the block of pixel values are transformed in a deterministic manner. Moreover, since DCT coefficients are quantized by a predefined quantization table in JPEG standard, the introduced error is also deterministic. In this sense, given the same input image and compression ratio, the same visual artifact will be obtained.

After inverse discrete cosine transform (IDCT), a quantization error in one DCT coefficient may affect every pixel in the 8×8 block. In other words, once a pixel is error-contaminated, its local neighborhood is very likely to be contaminated as well. This observation suggests us to model the *contaminated image* as Markov Random Field (MRF) [Li95]. MRF is the 2D expansion of Markov Random Chain. It has been widely used in image processing and analysis. In MRF, the probability that a pixel takes certain value is determined by the values of its local neighbors. This local property is known as Markovianity. As contaminated image

is subdivided into blocks of 8×8 pixels and each block is encoded independently, the visual artifact must also be localized. Hence it conforms to the spirit of MRF.

Our method is different from traditional MRF techniques which are usually iterative. Instead, our work is inspired by the previous work in texture synthesis [WL00, EF01]. Efros *et al.* [EL99] first modeled the sample input texture as MRF and synthesized larger texture seamlessly. The key idea is to synthesize a pixel value $\hat{q}$ by looking up the pixel value $\hat{p}$ in the sample texture with $\hat{p}$'s neighborhood matches $\hat{q}$'s neighborhood. Hertzmann *et al.* [HJO*01] extended the idea to a training-based technique for non-photorealistic rendering. Mese [MV01] proposed a Look Up Table (LUT) based method for inverse halftoning of images. Recently, Freeman [FJP02] proposed another training-based method for super-resolution. All these methods are based on MRF.

Intuitively speaking, the basic idea of our method is to learn the relationship between the original error-free (A) and the error-contaminated image (A') from a training set (Fig. 2). The training set contains pairs of the original and contaminated images. Having learned this knowledge, we *synthesize* an artifact-reduced image (B) instead of post-filter the artifact in the contaminated image (B'). The synthesis is performed in a pixel-by-pixel manner. When synthesizing a pixel $\hat{q}$ in image B based on the contaminated image B', the corresponding local neighborhood ω_q in B' is used for searching within the training set. When the best matching block ω is found in the contaminated image A', its corresponding center pixel $\hat{p}$ in the original image A is copied to the synthesized image B. This process is applied to every pixel in the edge region where the ringing artifact usually appears. Since we learn how artifact is introduced in the examples from the training set, we called our method a *training-based artifact-removal algorithm*.

3. Our Training-Based Method

3.1. Formation of Artifact Vectors

The first step is to setup a training set. We use several pairs of original images and their corresponding BDCT-encoded images (contaminated images) to form the training set. One way to obtain the training set is to randomly select images from any image library. However, a huge number of images are needed in order to acquire sufficient amount of artifact patterns.

Instead, we generate the training set to tailor for our purpose. As ringing artifact appears near sharp edges, we prepare the training set by generating images containing two basic shapes, *circle* and *corner* (Fig. 3). Although a circle looks simple, it captures most edge information. Almost any curve can be formed by connecting pieces of circular edges. The corner type images are mainly used to tackle sharp changing outlines like the star shape. Multiple instances of these two basic images are generated, each with different intensity, background intensity and line width.

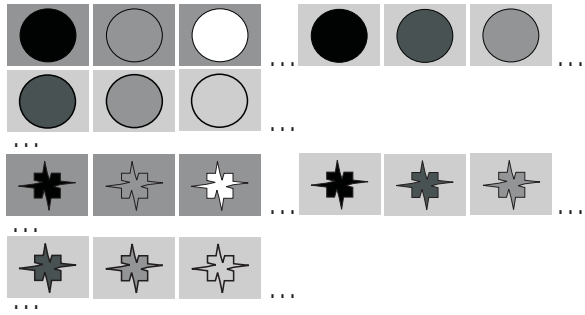


Figure 3: *Our training set.*

The most computational expensive part of our method is the searching of matching neighborhood. Naively searching the best matching block in the training set is impractical, especially when the size of the training set is large. Our approach is to extract numbers of *artifact patterns* from the training set and we only search within these artifact patterns. An artifact pattern is a square block of pixels (neighborhood), ω in the contaminated image A'. Each artifact pattern *implies* a corresponding center pixel value $\hat{p}$ in the original counterpart A (Fig. 2). Since the artifact pattern is reordered and stored in a vector form, we usually called it an artifact vector. From now on, we shall use the two terms, artifact pattern and artifact vector, interchangeably.

Selecting an appropriate size of artifact vector (the size of neighborhood) is a trade-off between the computational cost and the accuracy. Larger neighborhood in general provides more local information and hence returns better result, but the computation is more expensive. Since 8×8 blocks are used in BDCT, the size of neighborhood should not exceed 8×8. From our experience, a 5×5 neighborhood provides adequate local information and tractable computation. Since the ringing artifact mainly appears at region with strong edges, we only need to extract distinct artifact patterns from edge region. This can drastically reduce the number of distinct artifact patterns. Therefore, the first step is to identify the edge region. The identification process is shown in Fig. 4. We first apply an edge detection to the *contaminated* images (Fig. 4(a)) in the training set and edge pixels are marked (Fig. 4(b)). The reason we did not apply the edge detection to the original image is because the original image is not available during artifact removal. Next, we check each non-overlapping 8×8 block in the BDCT-encoded image to see if it contains any edge pixel. If so, the corresponding 8×8 block in the *original* image is tagged as *edge block* (Fig. 4(c)).

For each pixel $\hat{p}$ in the edge block of the *original* image, we look up its corresponding 5×5 neighborhood ω in the *contaminated* counterpart (Figure 5). This neighborhood is

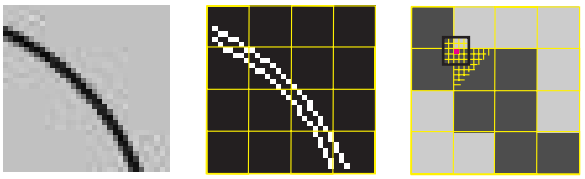


Figure 4: *Extraction of artifact vectors.*

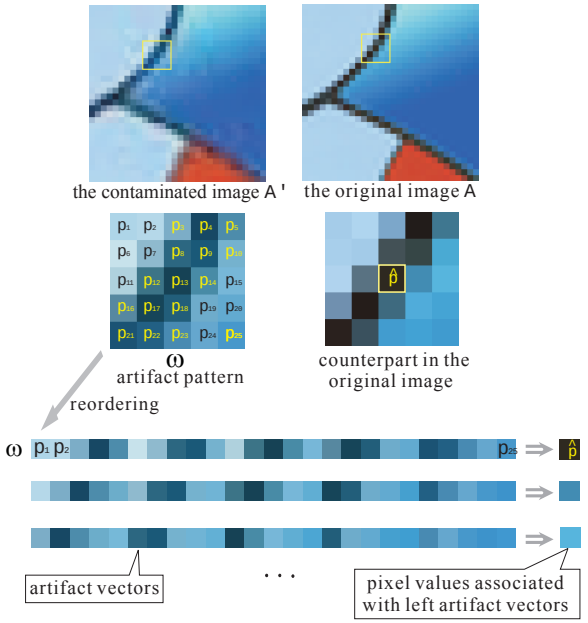


Figure 5: *Formation of artifact vectors.*

then linearized to form the artifact vector. If the artifact vector is distinct, it is stored together with $\hat{p}$. We say ω implies $\hat{p}$, and denoted as

$$\omega \Rightarrow \hat{p}\,.$$

Note that only ω is searched and matched during synthesis. The implied pixel value $\hat{p}$ is the data to retrieve. During the extraction of artifact vectors, the pixels near the image boundary may not be able to form artifact vectors due to the clipping by the image boundary.

3.2. Color Processing and Luminance Remapping

So far, we haven't mentioned how we handle the multiple color channels in color images. We did not perform artifact removal in *RGB* color space. Instead the color images are processed in *YUV* color space. As JPEG utilizes *YUV* color model in representing color images, processing in *YUV* color space avoids the overhead of color transformation. More importantly, the luminous component *Y* holds the human-sensitive visual information while chrominous components *U* and *V* hold the color differences which are less sensitive to human [Wan95]. This suggests that we can treat each component independently.

As illustrated in Fig. 6, we apply the proposed artifact-removal algorithm on the *Y* component only, while components *U* and *V* are retained. *U*, *V* components and processed *Y* component are then combined. We do not touch *U* and *V* components because the observable visual artifact is mainly due to the quantization error in *Y* component. This heterogeneous strategy reduces both memory consumption and computational cost due to searching, hence it shortens the time for artifact removal while retaining the visual quality of the recovered images. Since we only process the *Y* component, the training set only contains *Y* components of image pairs. The size of training set is hence reduced. To further generalize the extracted artifact patterns, we normalize them. This allows the algorithm to focus on the patterns rather than the absolute gray values. One way to normalize the pattern is to use histogram equalization. However, it is computational intensive, because we have to handle half million of artifact patterns. Instead, we apply the *luminance remapping* [HJO*01], a linear mapping that matches the means and variances of the luminance distributions. If p is the luminance of a pixel in the artifact pattern, and p' is the remapped luminance, luminance remapping is described by:

$$\frac{p-\mu_p}{\sigma_p} = \frac{p'-\mu_n}{\sigma_n}\,, \tag{1}$$

where μ_p and σ_p are the mean luminance and the standard deviation of the original pattern respectively; μ_n and σ_n are the mean luminance and standard deviation of the normalized pattern and predefined as 0.5 and 0.3 respectively. These normalized patterns are then stored. Note that this luminance remapping is also needed during the artifact removal.

3.3. Dominant Implication

Theoretically, one artifact pattern ω may not imply (be associated with) a unique pixel value $\hat{p}$. An artifact pattern may in fact imply a set of k possible pixel values, each with different probability.

$$\omega \Rightarrow \{\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_k\} \quad \text{and} \quad \sum_{i=1}^{k} \text{Prob}(\hat{p}_i) = 1\,,$$

where $\text{Prob}(\hat{p}_i)$ is the probability of $\hat{p}_i$.

The probability of each $\hat{p}_i$ can be obtained from the training set. During the extraction of artifact patterns, we can record the occurrences of each possible $\hat{p}_i$. The normalized count of each $\hat{p}_i$ will then be its probability. This probability distribution should be stored together with the artifact pattern. During the synthesis, the pixel value should be synthesized according to this probability distribution.

Interestingly, we observed a phenomenon that there is always a dominant pixel value $\hat{p}^*$ associated with each artifact

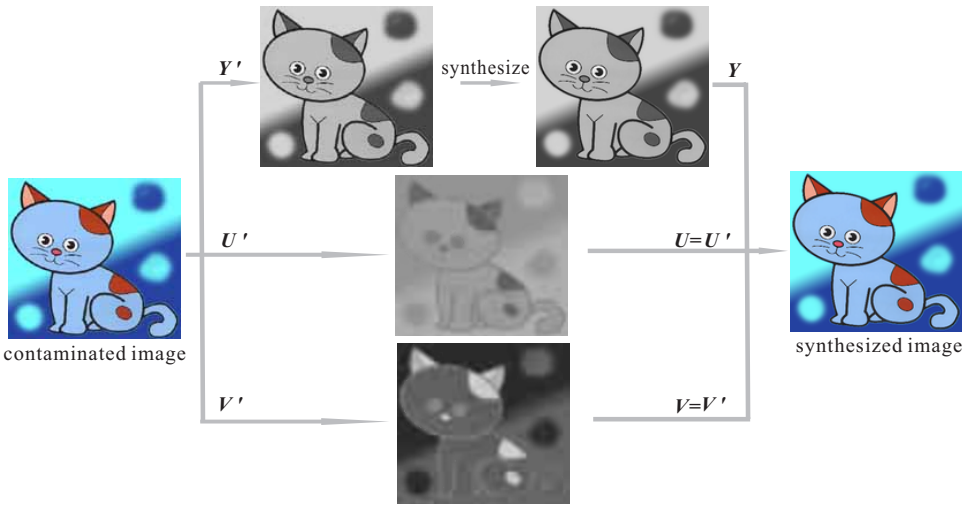


Figure 6: *Heterogeneous strategy for different color components. For visualization purpose, U and V components are shifted to $[0,1]$.*

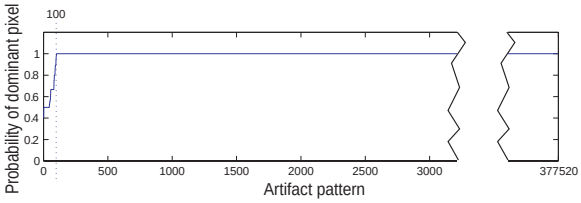


Figure 7: *Probability of dominant pixel.*

pattern. One reason is related to the deterministic properties of DCT and quantization. Another possible reason is that we focus on high-contrast images which usually show less variation in pixel values. High-contrast images usually have large smooth color regions and strong edges/outlines.

To verify our finding, we plot the probabilities of the dominant pixel values of all artifact patterns from our training set in Fig. 7. In this training set of 112 image pairs, there are 377,520 distinct artifact patterns. The average probability of dominant pixel value, associated with these patterns, is 99.98%. Among these artifact patterns, 377,420 patterns (99.97%) are associated with a unique pixel value. Only 100 patterns associate with two or more pixel values. In Fig. 7, we sort the artifact patterns according to the probabilities of their dominant pixel values, so that the smallest is on the left. The non-unique dominant pixels occupy a tiny portion only.

This observation of dominant implication suggests us a way to simplify and speed up the process. We can simply keep the dominant pixel value $\hat{p}^*$ and discard all other pixel values $\{\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_k\} - \{\hat{p}^*\}$. Hence, there is no need to store the probability distribution.

3.4. Tree-structured Vector Quantization

Even though we only extract artifact patterns from the Y component of color images, there are still enormous number of patterns from the training set. Management of such huge number of artifact patterns is crucial to fast query, hence fast synthesis. Naive searching among the sea of artifact patterns is impractical. Therefore, an indexing scheme is needed to achieve fast query. Furthermore, even though we have a lot of extracted artifact patterns, we may not find an exact match in some cases. In case of no exact matching, the indexing scheme should allow us to locate the closest artifact pattern.

To solve this problem, we employ the indexing technique called tree-structured vector quantization (TSVQ) [GG92]. TSVQ is usually used in data compression. Because of its hierarchical tree structure, an artifact pattern can be rapidly looked up. It also has a nice feature that allows efficient searching of the nearest pattern when the exact match does not exist.

The construction of TSVQ tree is as follows. Firstly, the centroid of all artifact vectors is computed, and this centroid represents all artifact vectors. A node holding this centroid is formed and is assigned as the root of TSVQ tree. Then we divide the vectors into τ_c groups, where τ_c is an user-defined constant. For each group, we compute the centroid and use it as the representative of that group. To do so, we create a node for each centroid and connect it as a child node of the root node. Each child node is actually the root of the subtree representing the corresponding group. Each group (subtree) is then recursively subdivided (branched) until one of the stopping criteria is reached.

To subdivide the set of artifact vectors into groups, we need a metric to measure the similarity among vectors. The

matching of pixels near the center pixel (the pixel being synthesized) is more important than that of pixels farther away. We employ the weighted Euclidean distance function.

$$D_e(\omega_a, \omega_b) = \sqrt{\sum_i \sum_j G(i,j)(\omega_a(i,j) - \omega_b(i,j))^2},$$

where ω_a and ω_b are artifact patterns being compared, and G is a 2D Gaussian kernel with higher weight at the center.

There are also two user-controllable stopping criteria: the number of subdivided groups τ_c, and the maximum tree depth τ_d. If any of them is satisfied, the subdivision should be stopped. Constant τ_c controls the number of groups after subdivision. That is, it controls the number of children of each interior node. If the number of vectors in the current group is less than τ_c, the subdivision should be stopped. Constant τ_d controls the maximum depth of the tree, excluding the root level. If the depth of the current group is equal to τ_d, further subdivision of the current branch should be prohibited.

3.5. Artifact Removal

Once the TSVQ tree is constructed, we can perform the artifact removal. The constructed TSVQ tree is actually the *priori* knowledge of BDCT-encoding artifact. Given a contaminated color image, we apply the synthesis on the Y component only as explained in Section 3.2.

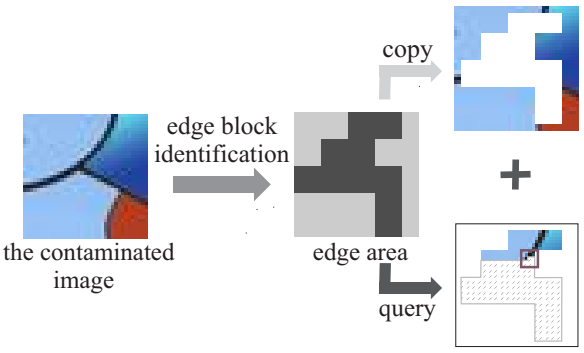


Figure 8: *Pixel classification.*

Just like the formation of artifact vectors, given an input contaminated image, we need to identify the edge and non-edge blocks. The same identification method as illustrated in Section 3.1 is used to identify edge blocks (Fig. 8). All pixels in the non-edge blocks of the contaminated image are simply copied to the output as illustrated on the upper path in Fig. 8. The non-edge blocks contain no recognizable strong edge and hence they rarely contain ringing artifact. On the other hand, all pixels in the edge blocks have to be synthesized by querying the TSVQ tree.

The query process is illustrated in Fig. 9. Given a contaminated image B', we synthesize an artifact-reduced image B in a pixel-by-pixel manner for each pixel in edge blocks. For

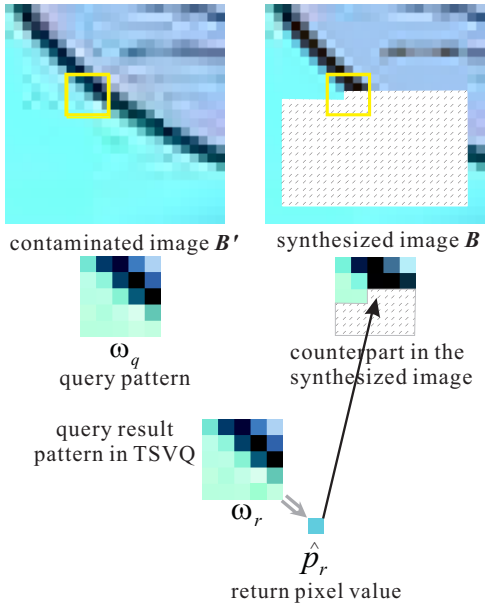


Figure 9: *Artifact removal.*

each pixel being synthesized, we form a query pattern ω_q by re-ordering its corresponding 5×5 neighborhood in image B'. The query vector is normalized using Equation 1 before query. The example contaminated image B' in Fig. 9 contains only edge blocks. Image B is initialized with a frame of width of 2 pixels. The reason is that pixels on this frame cannot form a 5×5 neighborhood for query. Therefore, these pixels are simply copied from the contaminated image B'. This is one restriction of our current approach.

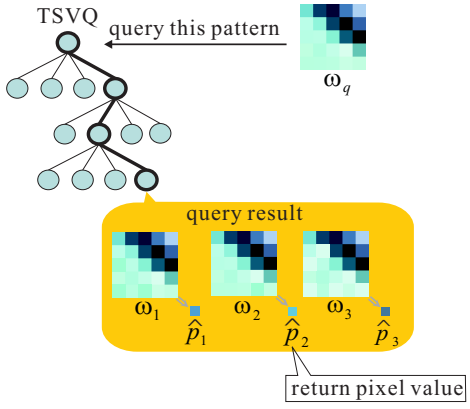


Figure 10: *The query process.*

We then query the artifact pattern ω_q, by traversing the TSVQ tree, as illustrated in Fig 10. Firstly, we compare the query vector ω_q with every child node of the root. The one with the smallest weighted Euclidean distance is selected.

The search continues to the next level of the branch until a leaf node is encountered. Since the closest pattern should be under this leaf node, we sequentially compare with all patterns ($\{\omega_1, \omega_2, \omega_3\}$ in this example). This sequential matching process does not take much time as the number of patterns inside a leaf node is very likely to be bounded by constant τ_c. In this example, ω_2 is the closest pattern. Its implied pixel value $\hat{p}_2$ is returned, luminance remapped (Equation 1), and copied to image *B*. The synthesis continues until the whole image *B* is filled.

4. Experimental Results

To verify the proposed method, we implement it and setup a training set. We create 112 images to setup such training set, as shown in Fig. 3. These images are originally not contaminated with BDCT compression. They are then compressed by the JPEG codec from Independent JPEG group to obtain 112 contaminated images. The TSVQ tree is constructed with $\tau_c = 9$ and $\tau_d = 7$, as we have found that this configuration maintains a good balance between the query efficiency and the complexity of tree.

To objectively measure the quality of synthesized images, we use peak signal-noise ratio (PSNR) and Structural SIMilarity (SSIM) [WBSS04]. PSNR is one of the most widely used image quality assessments. However, it does not fully capture the perceptual quality. Hence, we also measure our results using SSIM index which is a particular implementation of the philosophy of structural similarity. Larger SSIM value implies better image quality. The maximum SSIM value is 1.

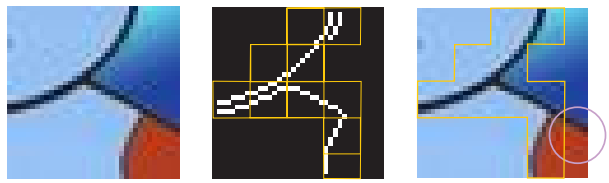
We test our method with 39 typical cartoon images. The control images are the original non-contaminated images. Our method obtains an average PSNR improvement of 0.76dB and average SSIM improvement of 0.022. The average time of synthesizing each image is 0.95s.

Fig. 11 visually compares three input contaminated images with the corresponding synthesized images. Parts of the images are blown up for comparison. The images in (a), (d) and (g) of Figure 11 are in the resolution of 200×232, 200×256, and 200×159 respectively. According to Table 1, our method improves in terms of both PSNR and SSIM.

5. Conclusions

By modeling the contaminated image as MRF, we propose a training-based method for reducing the ringing artifact in contaminated high-contrast images, such as cartoon images. Instead of post-filtering the contaminated image, we synthesize an artifact-reduced image. Using our method, the image quality of all test cases is substantially improved, not only subjectively but also objectively. Our method can effectively handle the ringing artifact which cannot be effectively solved by previous methods. Like other training-based methods, the size of training data affects the performance. In general, more training samples will give better PSNR improvement.

One limitation of our current method is illustrated in Fig. 12. In this example, the edge detection cannot detect the edge in the circle of Fig. 12(c). Therefore that particular block is tagged as non-edge block and no synthesis is performed. A partial solution is to make use of multiple edge detectors. However, no matter what kind of edge detection filter is used, it is still possible that some edges are not detected.



(a) (b) (c)

Figure 12: *Some edges (in the circle of (c)) may not be identified and hence cannot be synthesized.*

Currently, our method synthesizes images in a pixel-by-pixel manner. We believe the speed can be substantially increased if a patch-based approach is used. However whether the quality of synthesized images is still preserved requires further investigation.

References

[CHS98] CHAN Y.-H., HONG S.-W., SIU W.-C.: A practical postprocessing technique for real-time block-based coding system. *IEEE Transactions on Circuits and Systems for Video Technology 8*, 1 (Feb. 1998), 4–8.

[Com93] COMBETTES P. L.: The foundation of set theoretic estimation. *Proc. IEEE 81* (Feb. 1993), 182–208.

[CWQ01] CHEN T., WU H. R., QIU B.: Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts. *IEEE Transactions on Circuits and Systems for Video Technology 11*, 5 (May 2001), 594–602.

[EF01] EFROS A. A., FREEMAN W. T.: Image quilting for texture synthesis and transfer. In *SIGGRAPH 2001 Conference Proceedings* (2001), pp. 341–346.

[EL99] EFROS A. A., LEUNG T. K.: Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision* (Corfu, Greece, Sept. 1999), pp. 1033–1038.

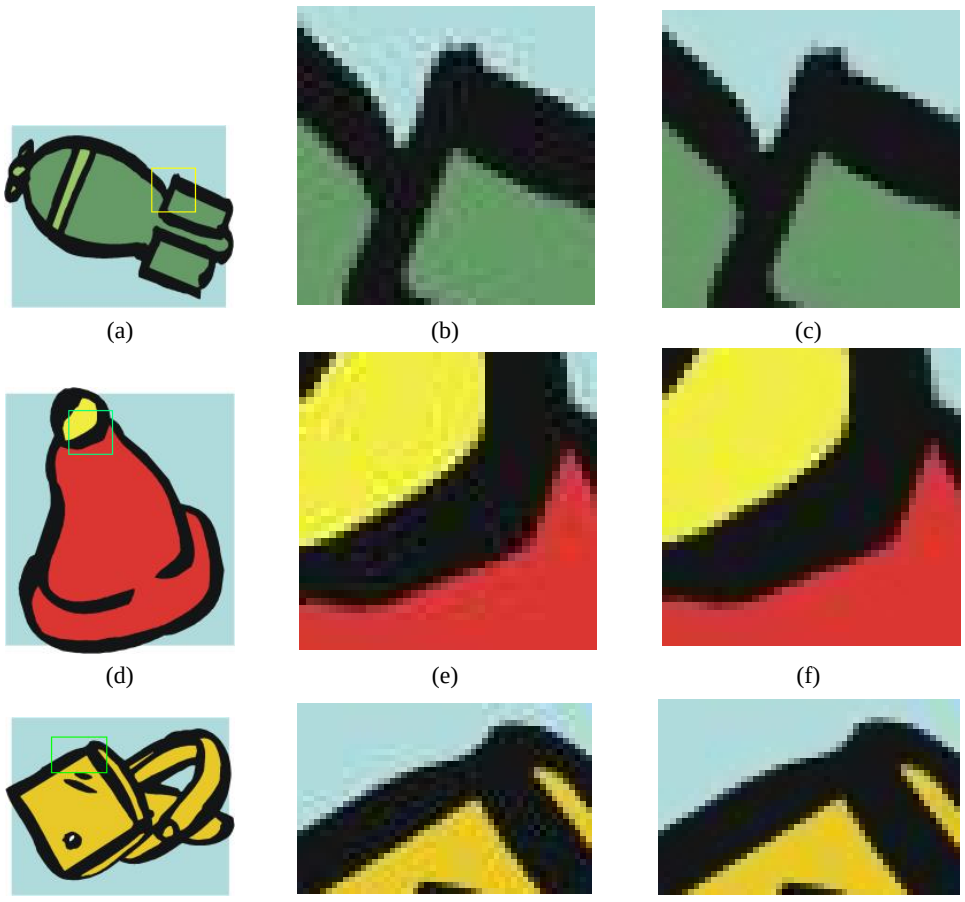


Figure 11: *Examples of artifact removal: bomb, hat, and bag. (a), (d) and (g) are the contaminated images. (b), (e) and (h) show the blow-up of the boxes in the corresponding contaminated images. (c), (f) and (i) blow up the corresponding parts in the synthesized images.*

Image	PSNR (dB)		SSIM		Time
	JPEG	Proposed	JPEG	Proposed	(s)
bomb	32.95	34.01	0.944	0.967	0.87
hat	31.84	32.35	0.922	0.936	0.87
bag	28.64	29.35	0.860	0.886	0.81

Table 1: *Statistical performance of our method.*

[FJP02] FREEMAN W. T., JONES T. R., PASZTOR E. C.: Example-based super-resolution. *IEEE Computer Graphics and Applications 22*, 2 (Mar./Apr. 2002), 56–65.

[GG92] GERSHO A., GRAY R. M.: *Vector quantization and signal compression*. Kluwer Academic Publishers, Boston, 1992.

[HJO*01] HERTZMANN A., JACOBS C., OLIVER N., CURLESS B., SALESIN D.: Image analogis. In *SIGGRAPH 2001 Conference Proceedings* (2001), pp. 327–340.

[JKK00] JEONG Y., KIM I., KANG H.: A practical

projection-based postprocessing of block-coded images with fast convergence rate. *IEEE Transactions on Circuits and Systems for Video Technology 10*, 4 (June 2000), 617–623.

[KH95] KUO C. J., HSIEH R. J.: Adaptive postprocessor for block encoded images. *IEEE Transactions on Circuits and Systems for Video Technology 5*, 4 (Aug. 1995), 298–304.

[LCPH96] LUO J., CHEN C. W., PARKER K. J., HUANG T. S.: Artifact reduction in low bit rate DCT-based image compression. *IEEE Transactions on Image Processing 5*, 9 (Sept. 1996), 1363–1368.

[Li95] LI S.: *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, New York, 1995.

[LKP98] LEE Y. L., KIM H. C., PARK H. W.: Blocking effect reduction of JPEG images by signal adaptive filtering. *IEEE Transactions on Image Processing 7*, 2 (Feb. 1998), 229–234.

[MV01] MESE M., VAIDYANATHAN P. P.: Look-up table (LUT) method for inverse halftoning. *IEEE Transactions on Image Processing 10*, 10 (Oct. 2001), 1566–1578.

[MZ95] MINAMI S., ZAKHOR A.: An optimization approach for removing blocking effects in transform coding. *IEEE Transactions on Circuits and Systems for Video Technology 5*, 2 (Apr. 1995), 74–82.

[PKL98] PAEK H., KIM R.-C., LEE S.-U.: On the pocs-based postprocessing techinque to reduce the blocking artifacts in transform coded images. *IEEE Transactions on Circuits and Systems for Video Technology 8*, 3 (June 1998), 358–367.

[PM93] PENNEBAKER W. B., MITCHELL J. L.: *JPEG Still Image Data Compression Standard*. Van Nostrand, New York, 1993.

[Qiu00] QIU G.: MLP for adaptive postprocessing block-coded images. *IEEE Transactions on Circuits and Systems for Video Technology 10*, 8 (Dec. 2000), 1450–1454.

[RH96] RAO K. R., HWANG J. J.: *Techniques and Standards for Image, Video, and Audio Coding*. Prentice Hall, NJ, 1996.

[RL84] REEVE H. C., LIM J. S.: Reduction of blocking effect in image coding. *Opt. Eng. 23*, 1 (Jan./Feb. 1984), 34–37.

[Wan95] WANDELL B.: *Foundations of Vision*. Sinauer Associates Inc., 1995.

[WBSS04] WANG Z., BOVIK A. C., SHEIKH H. R., SIMONCELLI E. P.: Image quality assessment: From error measurement to structural similarity. *IEEE Transactions on Image Processing 13*, 1 (Jan. 2004).

[WL00] WEI L. Y., LEVOY M.: Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH 2000 Conference Proceedings* (2000), pp. 479–488.

[WLY02] WEERASINGHE C., LIEW A. W., YAN H.: Artifact reduction in compressed images based on region homogeneity constraints using the projection onto convex sets algorithm. *IEEE Transactions on Circuits and Systems for Video Technology 12*, 10 (Oct. 2002), 891–897.

[YG97] YANG Y., GALATSANOS N. P.: Removal of compression artifacts using projections onto convex sets and line process modeling. *IEEE Transactions on Image Processing 6*, 10 (Oct. 1997), 1345–1357.

[Zak92] ZAKHOR A.: Iterative procedures for reduction of blocking effects in transform image coding. *IEEE Transactions on Circuits and Systems for Video Technology 2*, 1 (Mar. 1992), 91–95.