학습을 통한 JPEG 영상의 잡영 제거

김광인 O, 권영희, 김동호, 김진형
한국과학기술원 전산학과 인공지능연구실
{kimki O,kyhee,dkim,jkim}@ai.kaist.ac.kr

Learning to Remove JPEG Artifacts

Kwang In Kim O, Younghee Kwon, Dongho Kim, and Jin Hyung Kim
A. I. Lab, CS Dept., Korea Advanced Institute of Science and Technology

요 약

본 논문에서는 학습을 통해 JPEG 영상의 잡영을 제거하는 방법을 제안한다. 주어진 영상을 low-pass filtering 하면 JPEG 영상의 잡영 제거는 남아있는 저주파 성분으로부터 원래의 고주파 성분을 복원하는 문제가 된다. 다수의 영상과 이를 JPEG 압축한 영상의 쌍들이 주어졌을 때, low-pass filter 된 영상으로부터 원하는 고주파 성분을 추정하는 함수를 학습함으로써 JPEG 영상의 잡영을 제거하였다.

1. Introduction

Block-wise discrete cosine transform (DCT) coding has been successfully employed in image and video compression applications, including the Joint Photographers Expert Group (JPEG) image compression and Motion Pictures Expert Group (MPEG) video compression standards. The basic idea is to divide the image (or video frames) into disjoint blocks and then individually transform, quantize, and encode each block. This approach well exploits effectiveness of the DCT for the compression of small image blocks and its efficiency in the hardware implementation. However, at low bit rates, decoded images exhibit discontinuities that appear between the boundaries of the blocks (*block artifacts*) and oscillation in the vicinity of the major edges (*ringing artifacts*) [1]. Such artifacts can be reduced by low-pass filtering [1]. However, this introduces blurring in the reconstruction as high-frequency details of textured regions are smoothed out.

This paper approaches the problem of removing artifacts in JPEG compressed images by estimating missing high-frequency details from the low-pass filtered JPEG images (henceforth referred to as *smoothed images*) based on example pairs of original and smoothed images.

Actually there have been several approaches posing the JPEG artifact removal as the estimation or regression problem [1,2]. The main limitations of these methods are that they work based on the assumption of Gaussian-distributed data and are mostly linear. It is well known, however, that "interesting" structures in images such as edges or corners cannot be described by correlations [3,4], therefore, are not Gaussian. Furthermore, estimating high-frequency components from smoothed images are inherently nonlinear and accordingly, may not be modeled properly based on linear methods (Sec. 4).

In contrast to these methods, the proposed method does not make such an assumption of Gaussianity and approaches the problem with non-linear regression. The experimental results demonstrate the effectiveness of the proposed method.

2. Regression of high-frequency components of images

Let Y and X denote spaces of images and corresponding JPEG compressed images, respectively. After the band-pass filtering, the problem of block artifact removal is formulated as finding a map from the space of smoothed images to the corresponding target images:

$$f : LX \to Y ,$$

where L is the low-pass filter. The function f could be learned directly from example pairs of pixel based images $\{(Lx_1, y_1), \mathrm{K}, (Lx_l, y_l)\} \in LX \times Y$. However, more economical solution might be to estimate only the missing high-frequency components as the low-frequency counterparts are already provided in the input [5]. As a result, the target space is determined as the (missing) high-frequency components of the original images. This viewpoint also suggests representing the input images based on the band-frequency components[1] assuming conditional independency of the highest and lowest frequency components given band-frequency components [5].

As indicated by many other applications of example-based learning for related problems [6,7], proper representation of data is essential in the high-frequency estimation. This work proposes to represent the output high-frequency components based on the independent component analysis (ICA). The problem is then cast into learning a set of functions (regressions) from input band-frequency components to each independent component (IC; the representation of data in the ICA space) by which the output image is constructed as the linear combination of ICA basis. This way of representation is motivated by previous successful applications of ICA in image coding [8], denoising [9], and other related problems [10]. In these usual ICA applications, ICs of given data are computed simply by linearly projecting it to each ICA axis (i.e., taking the inner-product between the data and each normalized ICA axis vector). On the other hand, the proposed method estimates each IC based on a (possibly nonlinear) regression, which can be regarded as an extension of linear projection. In general, the ICA basis learned from general images was shown to be similar to wavelet basis the efficiency of which for representing images has well recognized in computer vision community especially when the object of interest is edges or bars [11] which are strongly related to high-frequency components.

3. Algorithm details

The FastICA algorithm [10] is utilized to obtain an output image representation $s \in S$ of $y \in Y$ ($h(Y) = S$). The advantages of the FastICA algorithm include a fast speed of convergence and orthogonality of the conversion matrix $\mathbf{B}$ (as a linear map, $h(y) = \mathbf{B}y$), which guarantees invertibility.

Before performing the ICA, a principal component analysis (PCA) is performed to whiten and reduce the dimensionality of the data. This reduces the number of scalar-valued functions ($f = \left[f^1, \mathrm{K}, f^r\right]^{\mathrm{T}}$) to be learned and enables avoiding estimation of the unnecessary complex image details. Since both the PCA and ICA are linear transforms, applying them can be represented based on a single $r \times M$ matrix $\mathbf{A}$, where r is the number of principal components (PCs) to be selected:

$$s = \mathbf{A}y .$$

[1] Actually, they correspond to the (remaining) high-frequency components of smoothed image; high-pass filtering is applied to extract them (Sec. 3).

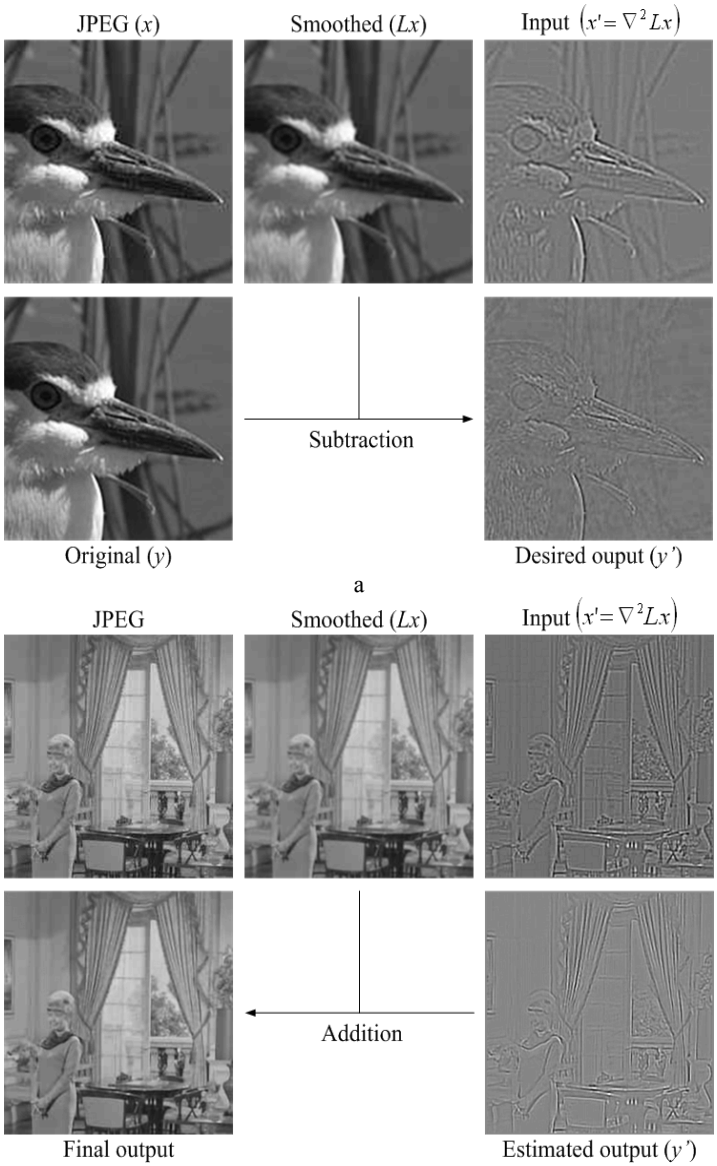

Fig. 1. Block diagrams of training (a) and testing (b) procedures: both sampling of training data and estimation are done in patch-wise manner.

Each IC s^i is then estimated based on a support vector machine (SVM) regression. The estimated s is then mapped back to the output space Y by inverting $\mathbf{A}$ to produce an image representation.

Up to this point, an image has been regarded as a point in a space whose dimensionality is equal to the number of pixels in the image. Yet this setting is rather restrictive since many applications of artifact removal require the capability of processing images with different sizes. Furthermore, from the viewpoint of learning functions, large image sizes can cause a serious overfitting problem.

Therefore, for dealing images with arbitrary sizes, a patch-based approach is adopted where a large image is regarded as a composition of patches (sub-windows of images) [5]. For this purpose, the input image is decomposed into a set of overlapping patches, where the corresponding target image patches are reconstructed independently. The final target image is then obtained by averaging the overlapping patches.

The training data for SVMs are then prepared by first JPEG-compressing a clean image y, then low-pass filtering the resulting JPEG image x (Lx). Thereafter, the desired output y' (as missing high-frequency components) is constructed by subtracting the smoothed image from the original image ($y' = y - Lx$). The input band-pass filtered data is prepared by taking the Laplacian of input image ($\nabla^2 Lx$).

To increase the efficiency of the training set, the data are contrast-normalized adopting the idea of [5]: during the construction of the training set, $\nabla^2 Lx$, and y' are all normalized based on dividing by $\|\nabla^2 Lx\|$. For an unseen image x , the input x' for f is prepared by computing $\nabla^2 Lx$ and dividing it by $\|\nabla^2 Lx\|$. The block artifact removal is then achieved by multiplying $\|\nabla^2 Lx\|$ with the target estimation $f(x')$ and adding the result to Lx . Fig. 1 summarizes the training and testing (removal of JPEG artifacts for unseen JPEG images) processes:

4. Experimental Results

For all experiments, the gray levels of images were normalized into [-1,1]. The training and testing image sets were disjoint. Fig. 2 shows examples of images used in training.[2]

Fig. 2. Example of training images.

For training, a set of smoothed and original image pairs was obtained by JPEG encoding and band-pass filtering original images. The training set was constructed by randomly sampling 15,000 image patches from these image pairs. Then, in the testing phase, the input JPEG image was smoothed as was done in the construction of training images. The PCA&ICA matrix $\mathbf{A}$ was obtained based on a sub-sample of size 10,000 by which the target high-resolution image representations for training SVMs are obtained.

Several hyper-parameters need to be tuned for SVMs, including the choice of kernel function, the regularization parameter, and the tolerance parameter for each estimator f^i of s^i [12]. While the best parameter determination depends on the problem of interest, they are optimized in the current work based on the cross-validation for each f^i . However, for the kernel function, a Gaussian kernel $k(x,y) = \exp\left(-\|x-y\|^2 / 2\sigma^2\right)$ with the same width parameter $\sigma > 0$ was used for all the estimators. Then, the cross validation was done by dividing the training set into distinct training and validation sets of size 10,000 and 5,000, respectively, and searching for the minimal validation error (for each IC) based on an interpolation search. After the parameter selection, each SVM was retrained based on all 15,000 training examples. The best σ was identified as 0.16 while the value of the regularization parameters varied from 0.2 to 1.6. Since no significant variation of errors observed with varying tolerance parameters they were fixed at 0.01. In addition to SVM-specific parameters, certain application-specific parameters were also tuned, including the number of ICs r and the input and output patch

[2] Some of the images used in the experiments were obtained from the USC-SIPI Image Database (http://sipi.usc.edu/database/).

sizes N and M. Choosing the patch size is a tradeoff between the risk of overfitting and the gain of contextual information. In this work, the input and output patch size were set at 9×9 and 5×5, respectively, based on cross validation on several training image pairs. Meanwhile, r was set at 15 which includes 99.74% of the energy from all 25 components. Fig. 3 shows examples of artifact removal. The 512×512-size Lena image and the 1024×1024-size Man image were used in the experiments. The JPEG images with low bit rate showed block and ringing artifacts which are clearly visible around the visor (Fig. 3c) and faces (Fig. 3d). Low-pass filtering removed these visually annoying artifacts. However this introduced severe blurring around eyes (Fig. 3e) and check (Fig. 3f). The proposed method successfully reconstructed sharp edges and at the same time did not produce such artifacts observed in Fig.3c and d. For comparison, reported performances of several existing methods for the same Lena image are summarized in Table 1.[3]

Table 1. Performance of different artifact removal methods for Lena image.

Artifact removal method	Improvement in PSNR
JPEG (PSNR=32.34dB)	
POCS [2]	0.45
Wavelet [13]	0.10
Reapplication of JPEG [14]	0.66
Proposed method	0.73

5. Discussions

This paper proposed a new method for block artifact removal in JPEG encoded images. After low-pass filtering the input JPEG images, the problem cast into estimating missing high-frequency components from given band-frequency components. The ICA was utilized in representing the target high-frequency components of images which were estimated based on SVMs. Experimental results showed the effectiveness of the proposed method.

There are various directions for future work. The proposed method, developed for removing JPEG artifacts, did not actually use any information on JPEG encoding scheme (except in the smoothing stage). This implies two consequences: 1. The proposed method could also be applied to different related problems: image denoising might be a good candidate; 2. There is room for improvement in JPEG block artifact removal: for example, the proposed method applied the same computation (smoothing and high-frequency estimation) evenly on the entire image. On the other hand, it has well-known that in textured or detailed regions blocking artifact is not noticeable due to the masking effect [1]. In these regions, it might be natural to reduce the effect of the low-pass filtering to maintain the details of images.

Acknowledgement. The ideas presented in this paper have greatly profited from discussions with Jahwan Kim and Se June Hong.

References

[1] K. Lee, D. S. Kim, and T. Kim, "Regression-based prediction for blocking artifact reduction in JPEG-Compressed Images," *IEEE Trans. IP*, vol. 14, no. 1, pp. 36-48, 2005.
[2] Y. Yang, N. Galatsanos, and A. Katsaggelos, "Projection-based spatially adaptive reconstruction of block-transform compressed images," *IEEE Trans. IP*, vol. 4, no. 7, pp. 896-908, 1995.
[3] P. Diaconis and D. Freeman, "On the statistics of vision: the Julesz conjecture," *J. Mathematical Psychology*, vol. 24, pp. 112-118, 1981.
[4] D. J. Field, "What is the goal of sensory coding?" *Neural Computation*, vol. 6, pp. 559-601, 1994.
[5] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *Int. J. Computer Vision*, vol. 40, no.1, pp. 25-47, 2000.
[6] O. D. Trier, A. K. Jain, and T. Taxt, "Feature extraction methods for character recognition-a survey," *Pattern Recognition*, vol. 29, no.4, pp. 641-662, 1996.
[7] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE. Trans. PAMI*, vol. 24, no. 1, pp. 34-58, 2002.
[8] J. H. van Hateren and A. van der Schaaf, "Independent component filters of natural images compared with simple cells in primary visual cortex," *Proc. R. Soc. Lond. B.*, vol. 265, pp. 359-366, 1997.
[9] E. Oja, A. Hyvärinen, and P. Hoyer, "Image feature extraction and denoising by sparse coding," *Pattern Analysis & Applications*, vol. 2, pp. 104-110, 1999.
[10] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, New York: John Wiley & Sons Ltd., 2001.
[11] A. Hyvärinen and E. Bingham, "Connection between multilayer perceptrons and regression using independent component analysis," *Neurocomputing*, vol. 50, pp. 211-222, 2003.
[12] V. Vapnik, *Statistical Learning Theory*, Wiley, 1998.
[13] Z. Xiong, M. Orchard, and Y. Zhang, "A Deblocking algorithm for JPEG compressed images using overcomplete wavelet representations," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 7, no. 2, pp. 433-437, 1997.
[14] A. Nosratinia, "Enhancement of JPEG-Compressed images by re-application of JPEG," *J. VLSI Signal Processing*, vol. 27, no. 1-2, pp. 69-79, 2001.

Fig. 3. Enlarged portions of image reconstructions: a and b. original images, c and d. JPEG encoded images, e and f. smoothed images, and g and h. results obtained with proposed method.

[3] PSNR values of JPEG encoded images were the same for all experiments. However, due to (possibly) different JPEG encoding scheme in each experiment, JPEG images could be slightly different from each other. Accordingly, the presented results should only be taken as baseline comparisons to facilitate the evaluation of the proposed method.