

# DEBLOCKING DCT-BASED COMPRESSED IMAGES WITH WEIGHTED TOTAL VARIATION

*François Alter, Sylvain Durand and Jacques Froment*

*François Alter* : ENS, 45 rue d'Ulm, 75005 Paris, France (*Francois.Alter@ens.fr*)

*Sylvain Durand* : LAMFA, université de Picardie, 33 rue Saint Leu,  
80039 Amiens cedex 1, France (*sdurand@cmla.ens-cachan.fr*)

*Jacques Froment* : LMAM, Université de Bretagne Sud, campus de Tohannic -  
Y. Coppens, BP 573, 56017 Vannes, France (*Jacques.Froment@univ-ubs.fr*)

## ABSTRACT

Coding systems based on block DCT, such as JPEG standard, are known to produce blocking and Gibbs effects. We propose a method to remove these artifacts without smoothing images and without losing their perceptual features. It consists in an weighted Total Variation minimization constrained by the knowledge of quantization intervals. A fast algorithm is proposed and experiments suggest better performance than state-of-the-art deblocking algorithms.

## 1. INTRODUCTION

JPEG and MPEG compressed images are currently in the million, and they remain still very popular despite the appearance of the new standard JPEG2000. However, they are known to contain, at low bit rate, annoying artifacts referred to blocking artifacts and Gibbs oscillation. JPEG, MPEG and some other compression standards partition images into  $8 \times 8$  (or  $16 \times 16$ ) blocks. A Discrete Cosine Transform (DCT) is applied to each block separately, prior to quantizing and coding. At reconstruction, images present discontinuities between adjacent blocks which are especially visible in smooth regions. Moreover, the cancellation of cosine coefficients in the vicinity of sharp edges generates Gibbs oscillations which are emphasized by the blocking structure.

Several postprocessing methods have been applied to remove these artifacts. They are based either on adaptive low-pass filtering, constrained least squares, projection onto convex sets (POCS), maximum a posteriori (MAP) estimations or more recently, diffusion [1, 2, 3, 4, 5] (see [6] for more references). In general, the main difficulty is to entirely remove blocking and Gibbs artifacts without smoothing edges. In [7], two of us proposed a method based on

Total Variation (TV) minimization, to remove Gibbs oscillations due to the thresholding of wavelet coefficients. The optimization is performed under the constraint that the significant coefficients remains unchanged. The TV was chosen for its relevance to regularize piecewise smooth images without smoothing edges [8]. As it was suggested in [7], this method can be adapted to quantization by a slight modification of the constraint. In the following, we adjust the model to reconstruct the block DCT coefficients that have been quantized by the compression process. The decompressed image will be, among all the images that fit the quantization interval, the one that has the smallest TV. We illustrate our method on JPEG compression, but the following may be easily adapted to other block DCT-based compression algorithms. TV is known to be very well suited for the removal of Gibbs phenomena. However, in order to remove blocking effect as well, we shall adapt the regularization functional to the blocks structure of JPEG images. The usual TV will be changed in a weighted TV that regularizes blocks' edges without regularizing images' true edges. The minimization is achieved by alternating between a subgradient descent and a projection on the constraint. Although the algorithm converges in infinite time, one obtains best PSNR with a very few number of iterations, leading therefore to a fast method.

TV regularization has already been used for the restoration of JPEG images. The more closely related method is Zhong's patent [9] implemented on a commercial software owned by Cognitech, Inc. Although we didn't personally conduct any experiment with this software, one has reported better results with our code than with the Zhong's one. It could be explained by the fact that the patent does not mention the use of an weighted-TV. Moreover, the TV functional seems to be lowered by directly acting to the level sets. The patent does not address the mathematical soundness of the algorithm. In [3], Whitaker et al use a weighted-TV regularization to restore JPEG images. But it is about a

---

This work was also supported in part by CNES and Science Pratique SA under Grant 762/00/CNES/8319; by CMLA/ENS Cachan, DOLabs and MAP5/Université Paris 5.

constrained least square leading therefore to a biased anisotropic diffusion scheme. Our model, as it relies on a projection, cannot be seen as a diffusion method in its usual sense. Notice also that the weight in [3] depends on the image, to the contrary of our weighted-TV functional.

## 2. WEIGHTED TOTAL VARIATION

In order to perform an efficient deblocking without removing the textures which are inside each block, the TV regularization should be stronger in the vicinity of blocks' edges. For that purpose, one designs a spatially adaptive functional by weighting the TV:

$$\text{TV}_\alpha(u) := \int_{\Omega} \alpha(x) |\nabla u| dx \quad (1)$$

The weight  $\alpha$  has to be positive, smooth and larger on block's edges. Weighted-TV have already been used in image processing, see e.g. [10][11][3]. The function  $\alpha$  is in general chosen so that it takes smaller values in textured areas and near edges while it takes larger ones in smooth areas, depending therefore in the image itself.

Here, the weight depends only on the blocks' size, needing no prior computation. It is defined directly on the discrete approximation of (1) following the  $8 \times 8$  block pattern:

$$J_\alpha(u) = \sum_{1 \leq m, n \leq N} \sqrt{a_{m,n}^2 + b_{m,n}^2 + c_{m,n}^2 + d_{m,n}^2} \quad (2)$$

with, denoting the modulo operator by  $\%$ ,

$$\begin{aligned} a_{m,n} &= \alpha_{m\%8} (u_{m+1,n} - u_{m,n}) \\ b_{m,n} &= \alpha_{n\%8} (u_{m,n+1} - u_{m,n}) \\ c_{m,n} &= \alpha_{(m-1)\%8} (u_{m,n} - u_{m-1,n}) \\ d_{m,n} &= \alpha_{(n-1)\%8} (u_{m,n} - u_{m,n-1}). \end{aligned} \quad (3)$$

$n$	0	1	2	3	4	5	6	7
$\alpha_n$	1.3	1.2	1.2	1.0	1.2	1.2	1.3	1.9

**Table 1.** Weight  $\alpha$  used for the weighted-TV.

## 3. CONSTRAINED TV MINIMIZATION

Denoting by  $T : \mathbf{R}^{N^2} \rightarrow \mathbf{R}^{N^2}$  the block DCT that associates to any image  $u$  of size  $N \times N$  the set of cosine coefficients  $T(u)$ , the constraint space  $U$  of images having the same quantized coefficients as the original image  $\tilde{u}$  will be

$$U = \left\{ u \in \mathbf{R}^{N^2} : \forall m, n = 1, \dots, N, \right. \\ \left. (T(u))_{m,n} \in [Q_{m,n}^-, Q_{m,n}^+] \right\} \quad (4)$$

where the input intervals  $([Q_{m,n}^-, Q_{m,n}^+])_{m,n}$  are given by the quantization table such that

$$\forall m, n = 1, \dots, N, (T(\tilde{u}))_{m,n} \in [Q_{m,n}^-, Q_{m,n}^+]. \quad (5)$$

The deblocked image  $u^*$  will be obtained by solving the following constrained optimization problem

$$\text{find } u^* \in U \text{ such that } J_\alpha(u^*) = \inf_{u \in U} J_\alpha(u) \quad (6)$$

Since  $J_\alpha$  is a convex function and  $U$  a convex set, solutions  $u^*$  are given by

$$\forall t > 0, u^* = P(u^* - tg(u^*)) \quad (7)$$

where  $P$  is the projector onto  $U$  that minimizes the distance and  $g(u)$  a subgradient of  $J_\alpha$  at  $u$ .

## 4. ALGORITHM

Equation (7) is numerically solved using a subgradient descent combining a projection on the constraint:

$$u^{k+1} = P(u^k - t_k g(u^k)) \quad (8)$$

where  $u^0 \in U$  is the decoded JPEG image. In a forthcoming long paper, we will prove that if the step  $t_k > 0$  is not decreasing too fast to 0 or, more precisely, if it satisfies

$$\sum_{k=0}^{+\infty} t_k = +\infty \text{ and } \sum_{k=0}^{+\infty} t_k^2 < +\infty, \quad (9)$$

then there exists  $u^*$  solving (6) such that  $\lim_{k \rightarrow +\infty} u^k = u^*$ .

This leads to the following algorithm where the only parameter is  $K$ . As  $K$  will remain small and independent of  $N$ , the overall complexity is of the same order as the block DCT and the inverse block DCT, that is  $O(N^2)$ :

1. get  $u_0$  by decoding the JPEG image; set  $k = 0$ ;
2. Choose e.g.  $t_k = 1/(k+1)$ .
3. Obtain  $g_k \in \partial J_\alpha(u_k)$ .
4. Set  $v^k = u^k - t_k g(u^k)$ .
5. Compute  $T(v^k)$ .
6. If  $(T(v^k))_{m,n} < Q_{m,n}^-$ , set  $P_{m,n} = Q_{m,n}^-$ ;  
if  $(T(v^k))_{m,n} > Q_{m,n}^+$ , set  $P_{m,n} = Q_{m,n}^+$ ;  
else, set  $P_{m,n} = T(v^k)_{m,n}$ .
7. Set  $u_{k+1} = T^{-1}(P)$
8. Set  $k = k + 1$ ; loop to 2 while  $k < K$ .
9. end : the decompressed image is  $u_K$ .

## 5. EXPERIMENTAL RESULTS

Using the tools given by the Independent JPEG Group's software, we have implemented the algorithm into the free and open-source MegaWave2 software [12]. We have experimented the code on various gray levels images, using both the plain-TV (that is, weight  $\alpha$  set to 1 in (3)) and the weighted-TV using the weights given by Table 1.

In order to converge to  $u^*$  one would like to choose  $K$  as large as possible, however a small value of  $K$  is recommended : the quality of the decompressed image does not grow when  $K$  is greater than a given threshold, and the Peak Signal to Noise Ratio (PSNR) may even slightly decreases. As we are no longer concerned with convergence and in order to reach the maximum at lower  $K$ , we now choose a constant step. On all tested images and for Compression Ratio (CR) below 30, the value  $t_k = 1/2$  allows to reach the maximum PSNR between 1 and 10 iterations, so that this value seems to be a good compromise between precision and speed. Also, experiments show that the corresponding optimal  $K$  grows linearly with the CR, so that it can be easily estimated by the decoder.

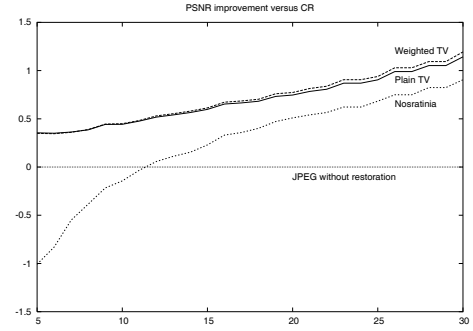
As a reference we have implemented Nosratinia's algorithm [4], a new and simple JPEG denoising method which re-applies JPEG compression to the shifted versions of the already compressed image, and computes the average. Despite its simplicity, following [4] this approach offers better performance than several other methods, including those based on nonlinear filtering, POCS and redundant wavelets. In Table 2, we add to the performance of deblocking algorithms given in [5] results associated to Nosratinia's, TV and weighted-TV algorithms. Together with Fig. 1, experimental results suggest that our proposed algorithm outperforms all other existing approaches with respect to the PSNR objective criterion. As a subjective criterion, Fig. 2 compares the images obtained with the weighted-TV and Nosratinia's algorithms. Our proposed weighted-TV algorithm offers better visual quality: Nosratinia's method introduces a slight blur, resulting in a loss of contrast near edges and of an unpleasant low-frequency pattern in smooth areas.

POCS	LPF	C-K	Nos	TV	WTV
0.29	0.82	0.87	0.62	0.87	0.91

**Table 2.** PSNR improvements on Lena by various deblocking methods, with JPEG PSNR = 31.24. First 3 columns are from [5] while last 3 columns correspond to Nosratinia's, proposed plain-TV and proposed weighted-TV algorithms.

## 6. REFERENCES

- [1] A. Zakhor, "Iterative procedures for reduction of blocking artifacts in transform image coding," *IEEE Trans. on Circuits and Systems for Video Tech.*, pp. 91–95, 1992.
- [2] Y. Yang, N.P. Galatsanos, and A. K. Katsaggelos, "Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 3, no. 6, pp. 421–432, 1993.



**Fig. 1.** Improvements of the PSNR versus CR on Lena.

- [3] R.T. Whitaker, A. Gothandaraman, and J. Gregor, "Total variation for the removal of blocking effects in dct based encoding," in *Proc. of ICIP'01*, 2001, pp. 455–458.
- [4] Aria Nosratinia, "Denoising of jpeg images by re-application of jpeg," *J. of VLSI Signal Processing*, vol. 27, pp. 69–79, 2001.
- [5] E. Choi and M.G. Kang, "Deblocking algorithm for dct-based compressed images using anisotropic diffusion," in *Proc. of ICASSP'03*, 2003, vol. 3, pp. 717–720.
- [6] M-Y. Shen and C-C. J. Kuo, "Review of postprocessing techniques for compression artifact removal," *J. Visual Comm. and Image Representation*, vol. 9, no. 1, pp. 2–14, 1998.
- [7] S. Durand and J. Froment, "Artifact free signal denoising with wavelets," in *Proc. of ICASSP'01*, 2001, vol. 6.
- [8] L.I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.
- [9] Sifen Zhong, "Image compression by optimal reconstruction," United States Patent 5,534,925, Cognitech Inc., July 1996.
- [10] D. Strong and T. Chan, "Relation of regularization parameter and scale in total variation based image denoising," UCLA CAM Report 96-7, University of California, Los Angeles, CA, 1996.
- [11] Lionel Moisan, "Extrapolation de spectre et variation totale pondérée," in *Proc. of Grets'01, Toulouse, France*, 2001.
- [12] J. Froment and L. Moisan (Eds), "Megawave2 v.2.20," A free and open-source Unix image processing software for reproducible research, available at <http://www.cmla.ens-cachan.fr>, 2003.



**Fig. 2.** Results on Lena. **A** : Original  $512 \times 512$ , 8 bpp image; **B** : zoom 4x of the original image; **line C-D-E** : JPEG compression at CR=15; **C** : standard decomposition (PSNR=33.72); **D** : weighted-TV algorithm (PSNR=34.34,  $K = 2$ , CPU=0.43 sec.); **E** : Nosratinia's algorithm (PSNR=33.95, CPU=6.15 sec.); **line F-G-H** : JPEG compression at CR=30; **F** : standard decomposition (PSNR=29.22); **G** : weighted-TV algorithm (PSNR=30.42,  $K = 10$ , CPU=1.19 sec.); **H** : Nosratinia's algorithm (PSNR=30.13, CPU=6.06 sec.). On printed paper it is not very easy to distinguish differences between weighted-TV and Nosratinia's restored images, but on a high resolution screen one can see that Nosratinia's images are slightly blurred. If the speed of Nosratinia's algorithm is about the same whatever the CR is, our algorithm goes slower when the CR grows since more iterations  $K$  are required. However, our method remains much faster than Nosratinia's one. CPU spent in second are given for a Pentium IV processor at 2.40Ghz, without any optimization and with identical DCT/IDCT software code.