ARTIFACT-FREE JPEG DECOMPRESSION WITH TOTAL GENERALIZED VARIATION

Kristian Bredies and Martin Holler

Institute of Mathematics and Scientific Computing, Karl-Franzens University Graz, Heinrichstraße 36, 8010 Graz, Austria

Keywords: Artifact-free JPEG Decompression, Total Generalized Variation, Image Reconstruction.

Abstract: We propose a new model for the improved reconstruction of JPEG (Joint Photographic Experts Group) images. Given a JPEG compressed image, our method first determines the set of possible source images and then specifically chooses one of these source images satisfying additional regularity properties. This is realized by employing the recently introduced *Total Generalized Variation (TGV)* as regularization term and solving a constrained minimization problem. In order to obtain an optimal solution numerically, we propose a primal-dual algorithm. We have developed a parallel implementation of this algorithm for the CPU and the GPU, using OpenMP and Nvidia's Cuda, respectively. Finally, experiments have been performed, confirming a good visual reconstruction quality as well as the suitability for real-time application.

1 INTRODUCTION

This paper presents a novel method for artifact-free reconstruction of given JPEG compressed images. Being a lossy compression, a given JPEG compressed object does not provide exact information about the original source image, but can be used to define a convex set of possible source images. Our method reconstructs an image in accordance with this given data and minimal total generalized variation (TGV) of second order. This recently introduced functional (Bredies et al., 2010) is well-suited for images as it is aware of both edges and smooth regions. In particular, its minimization discourages blocking and ringing artifacts which are typical for JPEG compressed images. It not only yields a significantly better approximation of the original image compared to standard decompression, but also outperforms, in terms of visual quality, existing similar variational approaches using different image models such as, for instance, the total variation.

The proposed model can be phrased as an infinite dimensional minimization problem which reads as

$$\min_{u \in L^2(\Omega)} \mathrm{TGV}_\alpha^2(u) + I_{U_D}(u) \qquad (1)$$

where TGV_α^2 is the total generalized variation functional of second order and I_{U_D} a convex indicator function corresponding to the given image data set U_D.

The reason for using the second order TGV functional is that for the moment it provides a good balance between achieved image quality and computational complexity. This is true especially since the improvement in the step from order one to order two, where order one corresponds to TV regularization, is visually most noticable (see (Bredies et al., 2010)). Generalizations to higher orders, however, seem to be possible and might lead to further improvements.

Among the continuous formulation, we also present a discretized model and an efficient solution strategy for the resulting finite-dimensional minimization problem. Moreover, we address and discuss computation times of CPU and GPU based parallel implementations.

Due to the high popularity of the JPEG standard, the development of improved reconstruction methods for JPEG compressed images is still an active research topic. In the following, we shortly address some of those methods. For a further review of current techniques we refer to (Nosratinia, 2001; Singh et al., 2007; Shen and Kuo, 1998). A classical approach is to apply filters, which only seems effective if space varying filters together with a pre-classification of image blocks is used. Another approach is to use algorithms based on projections onto convex sets (POCs), see for example (Kartalov et al., 2007; Weerasinghe et al., 2002; Zou and Yan, 2005), where one defines several convex sets according to data fidelity and regularization models. Typical difficulties of such methods are

the concrete implementation of the projections and its slow convergence.

Our approach, among others, can be classified as a constrained optimization method, where the constraints are defined using the available compressed JPEG data. One advantage of this method is that, in contrast to most filter based methods, we do not modify the image, but choose a different reconstruction in strict accordance to the set of possible source images. Since data fidelity is numerically realized by a projection in each iteration, we can ensure that at any iteration, the reconstructed image is at least as plausible as the standard JPEG reconstruction. Let us stress, however, that the idea of reconstructing JPEG images by minimizing a regularization functional under constraints given by the compressed data is not new. In (Bredies and Holler, 2011; Alter et al., 2005; Zhong, 1997) this was done using the well known total variation (TV) as regularization functional. In contrast to quadratic regularization terms, this functional is known to smooth the image while still preserving jump discontinuities such as sharp edges. But still, total variation regularized images typically suffer from the so called *staircasing effect* (Nikolova, 2000; Caselles et al., 2007; Ring, 2000), which limits the application of this method for realistic images.

The total generalized variation functional (TGV), introduced in (Bredies et al., 2010), does not suffer from this defect. As the name suggests, it can be seen as a generalization of the total variation functional: The functional may be defined for arbitrary order $k \in \mathbb{N}$, where in the case $k = 1$, it coincides with the total variation functional up to a constant. We will use the total generalized variation functional of order 2 as regularization term. As we will see, the application of this functional has the same advantages as the total variation functional in terms of edge preserving, with the staircasing effect being absent. Nevertheless, evaluation of the TGV functional of second order yields a minimization problem itself and nondifferentiability of both the TGV functional as well as the convex indicator function used for data fidelity make the numerical solution of our constrained optimization problem demanding.

The outline of the paper is as follows: In Section 2 we shortly explain the JPEG compression standard and introduce the TGV functional, in Section 3 we define the continuous and the discrete model and present a numerical solution strategy, in Section 4 we present experiments including computation times of CPU and GPU based implementations and in Section 5 we give a conclusion.
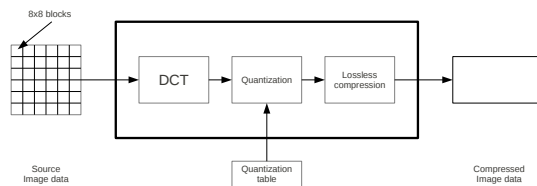

Figure 1: Schematic overview of the JPEG compression procedure, taken from (Bredies and Holler, 2011).

2 THEORETICAL BACKGROUND

2.1 The JPEG Standard

At first we give a short overview of the JPEG standard which is partly following the presentation in (Bredies and Holler, 2011). For a more detailed explanation we refer to (Wallace, 1991). The process of JPEG compression is lossy, which means that most of the compression is obtained by loss of data. As a consequence, the original image cannot be restored completely from the compressed object.

Let us for the moment only consider JPEG compression for grayscale images. Figure 1 illustrates the main steps of this process. At first, the image undergoes a blockwise discrete cosine transformation resulting in an equivalent representation of the images as the linear combination of different frequencies. This makes it easier to identify data with less importance to visual image quality such as high frequency variations. Next the image is quantized by pointwise division of each 8×8 block by a uniform quantization matrix. The quantized values are then rounded to integer, which is where the loss of data takes place, and after that these integer values are further compressed by lossless compression. The resulting data, together with the quantization matrix, is then stored in the JPEG object.
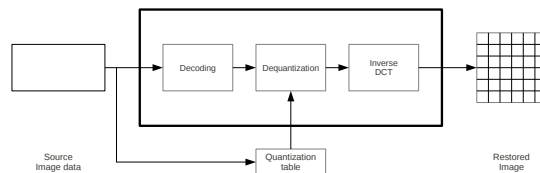

Figure 2: Schematic overview of the standard JPEG decompression procedure, taken from (Bredies and Holler, 2011).

The standard JPEG decompression, as shown in Figure 2, simply reverses this process without taking into account incompleteness of the data, i.e., that the compressed object delivers not a uniquely determined

Figure 3: JPEG image with typical blocking and ringing artifacts.

image, but a convex set of possible source images. Instead it just assumes the rounded integer value to be the true quantized DCT coefficient which leads to the well known JPEG artifacts as can be seen, for example, in Figure 3.

In the case of color images, typically a subsampling of color components is performed as part of JPEG compression. For that, color images are processed in the *YCbCr* color space, i.e., the images are given as $u = (y, cb, cr)$, where y is the luminance component and cb, cr are chroma components. Subsampling is then applied to the chroma components cb and cr, which reflects the fact that the human eye is less sensitive to color variations than to brightness variations. The resulting three image components, which now differ in resolution, then undergo the same process as grayscale images (see Figure 4). Again, the standard JPEG decompression simply reverts this process, now by applying an additional chroma-upsampling as last step.

An integral part of the model we propose is taking the set of all possible source images associated with a given JPEG object as a constraint. This set can be mathematically described as follows. With the integer coefficient data $(d_{i,j}^c)$, where the range of i, j depends on the resolution of the color component $c \in \{0, 1, 2\}$, and the quantization matrix $(Q_{i,j}^c)_{0 \leq i,j < 8}$, both provided by the compressed JPEG object, the set of numbers which yield $d_{i,j}^c$ in the quantization and rounding step is given by the interval

$$J_{i,j}^c = \left[Q_{i,j}^c \left(d_{i,j}^c - \frac{1}{2} \right), Q_{i,j}^c \left(d_{i,j}^c + \frac{1}{2} \right) \right]. \quad (2)$$

Note that for this notation we extend the quantization

matrices up to the image dimensions simply by repeating the 8×8 coefficients. Having this, the coefficient data set D can be defined as

$$D = \left\{ (z_{i,j}^c) \,|\, z_{i,j}^c \in J_{i,j}^c \text{ for all } c, i, j \right\} \quad (3)$$

and the set of possible source images of the compression process as

$$U_D = \left\{ u = (u_{i,j}^c) \,|\, ASu \in D \right\} \quad (4)$$

where A is a color-component-wise blockwise DCT operator and S is a subsampling operator (see Subsection 3.2 for the definition of A and S).

2.2 The TGV Functional

Another building block of the model is the total generalized variation functional (TGV) as proposed in (Bredies et al., 2010), in particular of second order. We give here a short definition and sum up some important results, for details and proofs however, we refer to (Bredies et al., 2010; Bredies and Valkonen, 2011). The pre-dual formulation of second-order TGV is given by

$$\mathrm{TGV}_\alpha^2(u) = \sup \left\{ \int_\Omega u \operatorname{div}^2 v \, \mathrm{d}x \,\middle|\, v \in C_c^2(\Omega, \mathcal{S}^{2 \times 2}), \right.$$

$$\left. \|v\|_\infty \leq \alpha_0, \|\operatorname{div} v\|_\infty \leq \alpha_1 \right\}, \quad (5)$$

where $\alpha = (\alpha_0, \alpha_1) \in \mathbb{R}^2$, $u \in L^1(\Omega)$ and $\mathcal{S}^{2 \times 2}$ is the space of symmetric 2 by 2 matrices. Given a function $u \in L^1(\Omega)$, the second-order TGV functional takes into account, as the test functions have the form $\operatorname{div}^2 v$, the generalized derivative of u up to order 2. Its kernel is the set of polynomials with degree less than 2. Evaluation of TGV_α^2 can be also interpreted as an optimal balancing of the first and second generalized derivative of u among themselves. This becomes obvious, considering an equivalent representation of TGV_α^2:

$$\mathrm{TGV}_\alpha^2(u) = \inf_{v \in \mathrm{BD}(\Omega, \mathbb{R}^d)} \alpha_1 \|\nabla u - v\|_1 + \alpha_0 \|\mathcal{E}(v)\|_1. \quad (6)$$

Here, $\mathcal{E}(v)$ denotes the symmetrized derivative of the vector field v and $\mathrm{BD}(\Omega)$ the space of vector fields of bounded deformation. As one can also see in (6), the ratio between the parameters α_1 and α_0 weights the balancing between the first and second derivative of u and will later also influence possible solutions of TGV_α^2 regularized optimization problems. The TGV functional possesses several properties which support its usage as regularization term, such as convexity and lower semi-continuity with respect to L^1 convergence. It also satisfies a Poincaré-Wirtinger type inequality which can be used to obtain coercivitiy of the objective functional.
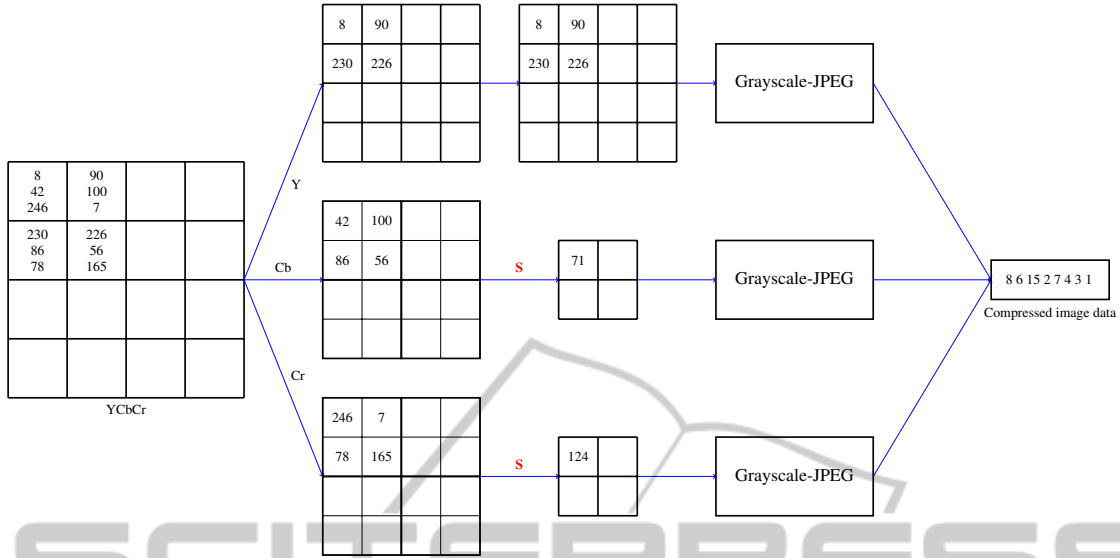
Figure 4: Chroma subsampling during JPEG compression.

3 THE METHOD

This section is devoted to present the TGV-based JPEG reconstruction method. Let us at first state the associated minimization problem in an infinite dimensional setting. Note that we abuse notation by using the same symbols for the continuous setting as for the discrete one.

3.1 The Continuous Setting

For convenience we only consider grayscale images for the continuous setting. Hence, our images are represented by real valued functions $u \in L^2(\Omega)$ with Ω a bounded Lipschitz domain, typically a rectangle. Using the definition of the TGV functional given in (5) we can formulate the infinite dimensional minimization problem as follows:

$$\min_{u \in L^2(\Omega)} \mathrm{TGV}_\alpha^2(u) + I_{U_D}(u) \quad (7)$$

where

$$I_{U_D}(u) = \begin{cases} 0 & \text{if } u \in U_D, \\ \infty & \text{else.} \end{cases}$$

The data set U_D can be defined as

$$U_D = \{u \in L^2(\Omega) \,|\, Au \in D\}$$

where A is a basis transformation operator related to a general orthonormal basis $(a_n)_{n \geq 0}$ of $L^2(\Omega)$. The coefficient data set $D \subset \ell^2$ reflects interval restrictions on the coefficients:

$$D = \{z \in \ell^2 \,|\, z_i \in J_i \,\forall i \in \mathbb{N}_0\}$$

where $J_i \subset \mathbb{R}$ is a non-empty, closed (not necessarily bounded) interval for any $i \in \mathbb{N}_0$. Note that this model allows arbitrary orthonormal bases $(a_n)_{n \geq 0}$ of $L^2(\Omega)$ and constraints on infinitely many coefficients. However, for JPEG decompression, one usually chooses an infinite block-cosine orthonormal basis, and lets only the J_i associated with the first 8×8 blockwise coefficients be bounded and $J_i = \mathbb{R}$ for the remaining i, see (Bredies and Holler, 2011) for details.

Under assumptions which are satisfied in the application of this model to JPEG decompression, we can show existence of a solution to (7). More generally, if we assume that U_D has non-empty interior and that $J_{n_0}, J_{n_1}, J_{n_2}$ are bounded for certain n_0, n_1, n_2, existence of a solution can be guaranteed.

3.2 The Discrete Model

Based on the equivalent formulation of the TGV functional in (6) and the data set U_D as in (4) we will now formulate our discrete model.

For $k, l \in \mathbb{N}$, we set the space of discrete color images $U = \mathbb{R}^{8k \times 8l \times 3}$ and further $V = U^2$ and $W = U^3$. The dimension $8k \times 8l \times 3$ results from the three color components and reflects the fact that any JPEG image is processed with its horizontal and vertical number of pixels being multiples of 8.

Now for $u \in U$, a discrete vector-input version of the TGV_α^2 functional can be defined as

$$\mathrm{TGV}_\alpha^2(u) = \inf_{v \in V} \alpha_1 \|\nabla(u) - v\|_V + \alpha_0 \|\mathcal{E}(v)\|_W, \quad (8)$$

where $\nabla : U \rightarrow V$ denotes a discrete, color-component-wise gradient operator using forward differences and $\mathcal{E} : V \rightarrow W$ denotes a discrete, color-component-wise symmetric gradient operator using backward differences, i.e., $\mathcal{E}(v) = \frac{1}{2}(J(v) + J(v)^T)$ with $J(v)$ a discrete color-component-wise Jacobian of v. Note that in $\mathcal{E}(v)$, the off-diagonal entries need to be stored only once, thus $\mathcal{E}(v) \in U^3$. For $v = (v_{i,j})_{0 \leq i,j < 8k}$ with $v_{i,j} \in \mathbb{R}^{3 \times 2}$ the norm on V is defined as

$$\|v\|_V := \sum_{i,j} |v_{i,j}|$$

with $|\cdot|$ the Frobenius norm on $\mathbb{R}^{3 \times 2}$. The norm $\|\cdot\|_W$ is defined similarly.

In order to avoid extensive indexing, we will now give just a local, component-wise definition of the operators S and A, necessary to describe the data set U_D. The subsampling operator S depends on the foregoing chroma subsampling process, but typically is defined on disjoint 2×2 blocks of each chroma component, denoted by $(z_{i,j})_{0 \leq i \leq 1}$, as

$$Sz = \frac{1}{4} \sum_{m,n=0}^{1} z_{m,n},$$

reducing the resolution of the chroma components by the factor 4. Since the resolution of the brightness component is not reduced, S is the identity for this component. The discrete cosine transformation operator is defined, for each color component, on each disjoint 8×8 block $(z_{i,j})_{0 \leq i,j \leq 7}$, as

$$(Az)_{p,q} = C_p C_q \sum_{n,m=0}^{7} z_{n,m} \cos\left(\frac{\pi(2n+1)p}{16}\right)$$
$$\cos\left(\frac{\pi(2m+1)q}{16}\right),$$

for $0 \leq p,q \leq 7$ and

$$C_s = \begin{cases} \frac{1}{\sqrt{8}} & \text{if } s = 0, \\ \frac{1}{2} & \text{if } 1 \leq s \leq 7. \end{cases}$$

With the operators S and A, the set U_D can now be defined as already done before in (4) by

$$U_D = \{u \in U \,|\, ASu \in D\},$$

where the coefficient data set D is obtained from the compressed JPEG object as in (3).

With these prerequisites, the finite dimensional optimization problem for artifact-free JPEG decompression reads as

$$\min_{u \in U} \mathrm{TGV}_\alpha^2(u) + I_{U_D}(u), \qquad (9)$$

where again

$$I_{U_D}(u) = \begin{cases} 0 & \text{if } u \in U_D, \\ \infty & \text{else.} \end{cases}$$

Using the boundedness of the data intervals $J_{i,j}^c$ defined in (2) it can be shown that there exists a solution to (9) and that this problem is equivalent to

$$\min_{(u,v) \in U \times V} F(K(u,v)) + I_{U_d}(u), \qquad (10)$$

where $K : U \times V \rightarrow V \times W$,

$$K = \begin{bmatrix} \nabla & -1 \\ 0 & \mathcal{E} \end{bmatrix}$$

and $F : V \times W \rightarrow \mathbb{R}$,

$$F(v,w) = \alpha_1 \|v\|_V + \alpha_0 \|w\|_W.$$

This formulation will now be the basis for the numerical approach.

3.3 A Primal-dual Algorithm

We numerically solve our minimization problem using a primal-dual algorithm presented in (Chambolle and Pock, 2011), for which convergence can be ensured. This algorithm is well-suited for this problem because, as we will see, all necessary steps during one iteration reduce to simple arithmetic operations and the evaluation of a forward and inverse block-cosine transformation, for which highly optimized code already exists. This makes the algorithm fast and also easy to implement on the GPU.

As first step we note that (10) is equivalent to the following saddle-point problem:

$$\min_{x \in X} \max_{y \in Y} \left((y, Kx)_{Y,Y} - F^*(y) + I_{U_D}(x) \right) \qquad (11)$$

where $X = U \times V$, $Y = V \times W$, $(\cdot, \cdot)$ is the scalar product on Y and F^* is the convex conjugate of F. The primal-dual strategy for finding saddle points presented in (Chambolle and Pock, 2011) amounts to performing the abstract iteration shown in Algorithm 1. Note that ∂F^* and ∂I_{U_D} refers to the subdifferential of F^* and I_{U_D}, respectively, and the operator K^* denotes the adjoint of K and is given by

$$K^* = \begin{bmatrix} -\operatorname{div} & 0 \\ -1 & -\operatorname{div}_2 \end{bmatrix}$$

with $\operatorname{div} = -\nabla^*$ and $\operatorname{div}_2 = -\mathcal{E}^*$ denoting discrete divergence operators.

Using standard arguments from convex analysis, it can be shown that the resolvent-type operators $(I + \sigma \partial F^*)^{-1}$ and $(I + \tau \partial I_{U_D})^{-1}$ take the following form:

$$(I + \sigma \partial F^*)^{-1}(v,w) = \left(\operatorname{proj}_{\alpha_1}(v), \operatorname{proj}_{\alpha_0}(w)\right)$$

where

$$\text{proj}_{\alpha_1}(v) = \frac{v}{\max(1, \frac{\|v\|_\infty}{\alpha_1})},$$

$$\text{proj}_{\alpha_0}(w) = \frac{w}{\max(1, \frac{\|w\|_\infty}{\alpha_0})}$$

and

$$(I + \tau \partial I_{U_D})^{-1}(u, v) = \left(u + S^{-1}\left(\text{proj}_{U_A}(Su) - Su\right), v\right)$$

where

$$\text{proj}_{U_A}(u) = A^* z$$

with

$$z_{i,j}^c = \begin{cases} u_{i,j}^c & \text{if } (Au)_{i,j}^c \in J_{i,j}^c = [l_{i,j}^c, r_{i,j}^c] \\ r_{i,j}^c & \text{if } (Au)_{i,j}^c > r_{i,j}^c \\ l_{i,j}^c & \text{if } (Au)_{i,j}^c < l_{i,j}^c, \end{cases}$$

$A^* = A^{-1}$ the adjoint of A and S^{-1} denoting the upsampling operator associated with S, given locally by replication of $z \in \mathbb{R}$, i.e.,

$$S^{-1} z = \begin{pmatrix} z & z \\ z & z \end{pmatrix}.$$

Having this, we can now give the concrete implementation of the primal-dual algorithm for JPEG decompression in Algorithm 2. Note that the step-size restriction $\sigma\tau \leq \frac{1}{12}$ results from the estimate $\|K\|^2 < 12$. As one can see, all steps of Algorithm 2 can be evaluated by simple, mostly pixel-wise operations making each iteration step fast.

4 NUMERICAL EXPERIMENTS

We implemented and tested the proposed TGV-based JPEG decompression method. First of all, we have compared the standard JPEG decompression with our method for three images possessing different characteristics. The outcome can bee seen in Figure 5, where also the image dimensions and the memory requirement of the JPEG compressed image is given in *bits*

Algorithm 1: Abstract primal-dual algorithm.

- Initialization: Choose $\tau, \sigma > 0$ such that $\|K\|^2 \tau\sigma < 1$, $(x^0, y^0) \in X \times Y$ and set $\bar{x}^0 = x^0$
- Iterations ($n \geq 0$): Update $x^n, y^n, \bar{x}^n$ as follows:

$$\begin{cases} y^{n+1} = & (I + \sigma\partial F^*)^{-1}(y^n + \sigma K\bar{x}^n) \\ x^{n+1} = & (I + \tau\partial I_{U_D})^{-1}(x^n - \tau K^* y^{n+1}) \\ \bar{x}^{n+1} = & 2x^{n+1} - x^n \end{cases}$$

---

per pixel (bpp). As one can see, our method performs well in reducing the typical JPEG artifacts and still preserves sharp edges.

Figure 6 then allows to compare our results with the reconstruction using TV instead of TGV as regularization functional as proposed in (Bredies and Holler, 2011). As one can see, in particular in the surface plots, the TV-based method also maintains sharp edges. However, it leads to a staircasing effect in regions that should be smooth. In contrast to that, the TGV-based method yields a more natural and visually more appealing reconstruction in such regions.

Figure 7 serves as an example of an image containing texture. It can be seen that our method preserves fine details and does not lead to additional smoothing of textured regions.

We also developed a parallel implementation of the method for multi-core CPUs and GPUs, using OpenMP (OpenMP Architecture Review Board, 2011) and Nvidia's Cuda (NVIDIA, 2008), respectively. For the GPU implementation we partly used kernel functions adapted to the compute capability of the device. The blockwise DCT was performed on the CPU and the GPU using FFTW (Frigo and Johnson, 2005) and a block-DCT kernel provided by the Cuda SDK, respectively. Computation times of those implementations for multiple image sizes are given in Table 1. As one can see, especially the GPU implementation yields a high acceleration and makes the method suitable for real-time applications. The given

---

Algorithm 2: Scheme of implementation.

1: **function** TGV-JPEG(J_{comp})
2:     $(d, Q) \leftarrow$ Decoding of JPEG-Object J_{comp}
3:     $d \leftarrow d \cdot Q$
4:     $u \leftarrow S^{-1}(A^*(d))$
5:     $v \leftarrow 0, \bar{u} \leftarrow u, \bar{v} \leftarrow 0, p \leftarrow 0, q \leftarrow 0$
6:     choose $\sigma, \tau > 0$ such that $\sigma\tau \leq 1/12$
7:     **repeat**
8:         $p \leftarrow \text{proj}_{\alpha_1}(p + \sigma(\nabla\bar{u} - \bar{v}))$
9:         $q \leftarrow \text{proj}_{\alpha_0}(q + \sigma(\mathcal{E}(\bar{v})))$
10:       $u_+ \leftarrow u + \tau(\text{div}\, p)$
11:       $v_+ \leftarrow v + \tau(p + \text{div}_2\, q)$
12:       $u_+ \leftarrow u_+ + S^{-1}(\text{proj}_{U_A}(Su_+) - Su_+)$
13:       $\bar{u} \leftarrow (2u_+ - u), \bar{v} \leftarrow (2v_+ - v)$
14:       $u \leftarrow u_+, v \leftarrow v_+$
15:     **until** Stopping criterion fulfilled
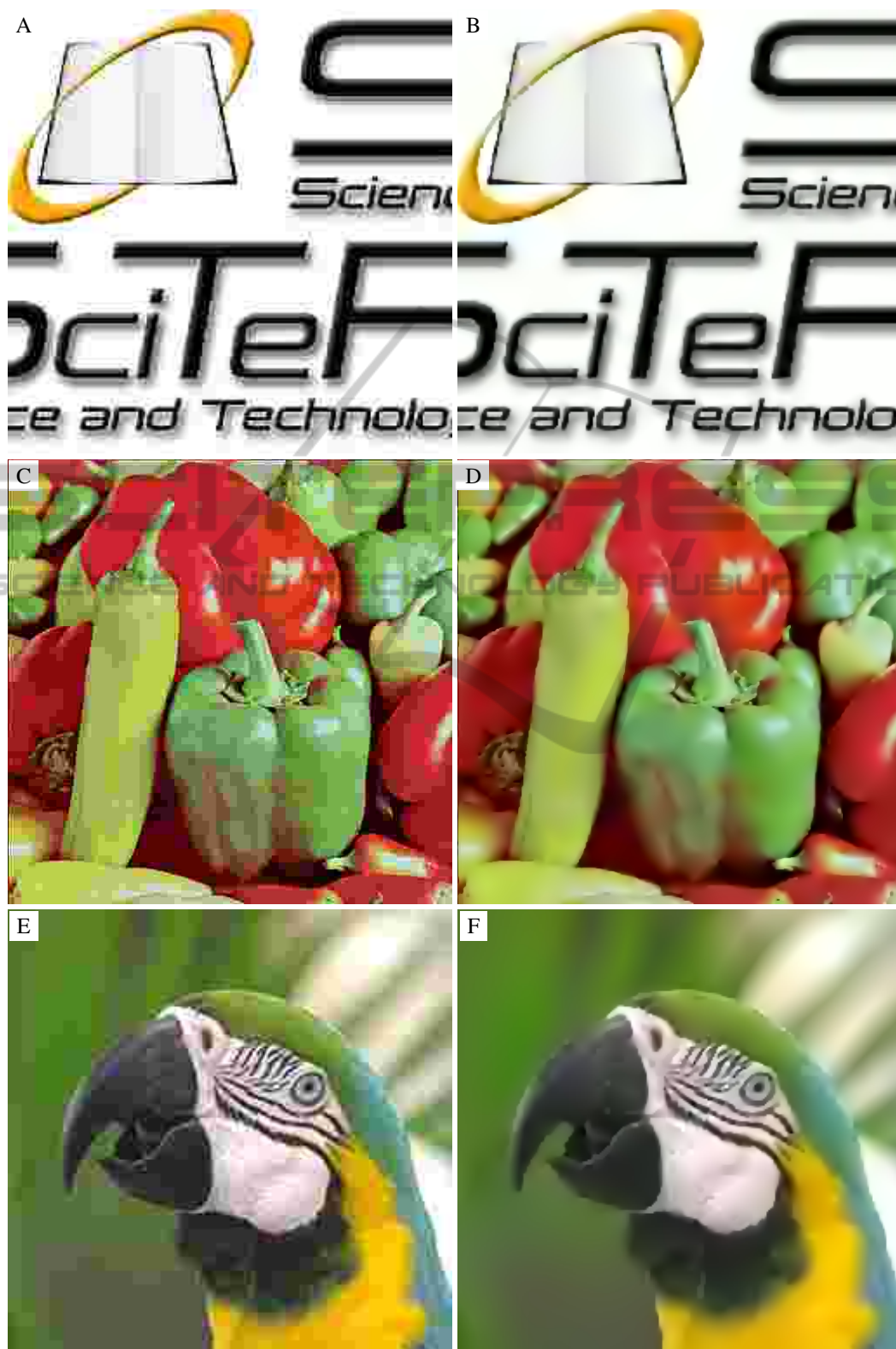16:   **return** u_+
17: **end function**

Figure 5: On the left: Standard decompression, on the right: TGV-based reconstruction after 1000 iterations. A-B: SciTePress image at 0.5 bpp (256 × 256 pixels). C-D: Peppers image at 0.15 bpp (512 × 512 pixels). E-F: Parrot image at 0.3 bpp (256 × 256 pixels).
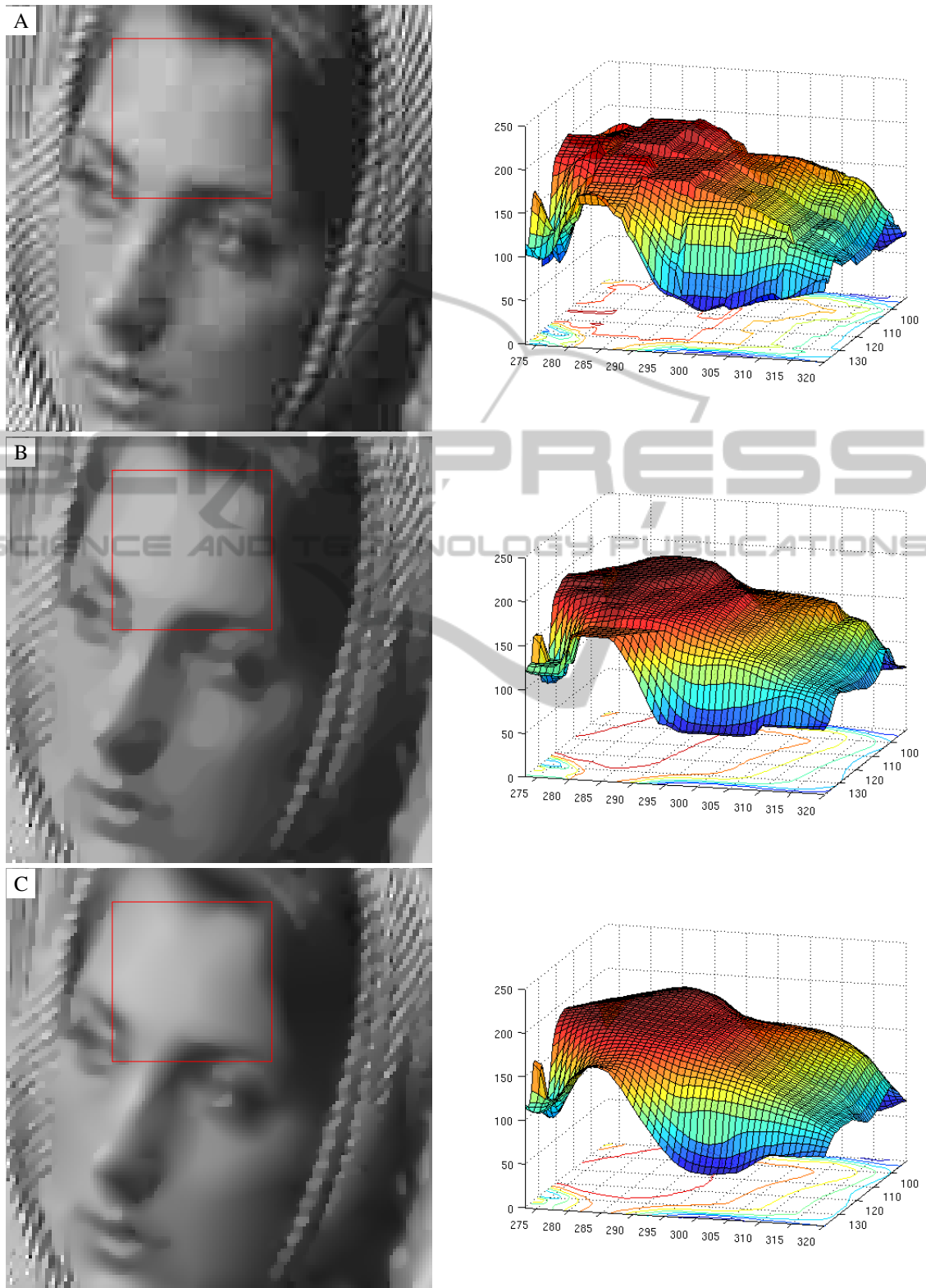
Figure 6: Close-up of Barbara image at 0.4 bpp at 1000 iterations. The marked region on the left is plotted as surface on the right. A: Standard decompression. B: TV-based reconstruction. C: TGV-based reconstruction.

Table 1: Computation times in seconds to perform 1000 iterations for different devices and image sizes. CPU: AMD Phenom 9950. GPUs: Nvidia Quadro FX 3700 (compute capability 1.1), Nvidia GTX 280 (compute capability 1.3), Nvidia GTX 580 (compute capability 2.0). Note that on the Quadro FX 3700 and GTX 280, not enough memory was available to perform the algorithm for the 3200×2400 pixel image.

Device	512×512	1600×1200	3200×2400
CPU Single-core	53.22	672.51	1613.44
CPU Quad-core	28.32	263.70	812.18
GPU Quadro FX 3700	4.92	35.52	-
GPU Nvidia GTX 280	2.2	10.22	-
GPU Nvidia GTX 580	1.2	6.6	25.70

Figure 7: From left to right: Uncompressed image at 24 bpp (256×256 pixels), standard decompression of JPEG compressed image at 1.06 bpp, TGV-based reconstruction of the same JPEG compressed image after 1000 iterations. Image by (Dawgbyte77, 2008), licensed under CC-BY-2.0 (http://creativecommons.org/licenses/by/2.0/).

computation times correspond to the computation of 1000 iterations, which is in most cases more than enough for a reconstruction visually almost indistinguishable from one obtained as optimal solution satisfying a suitable stopping criterion. Since the decrease of the TGV-value of the image is typically very high especially during the first iterations of the algorithm, and since a fit-to-data can be ensured for any iteration step image, one could use the image obtained after only a few number of iterations as intermediate reconstruction and then iteratively update the solution.

5 SUMMARY AND CONCLUSIONS

We proposed a novel method for the improved reconstruction of given JPEG compressed images, in particular, color images. The reconstruction is performed by solving a non-smooth constrained optimization problem where exact data fidelity is achieved by the usage of a convex indicator functional. The main novelty, however, lies in the utilization of TGV of sec-

ond order as a regularization term which intrinsically prefers natural-looking images, as it has been confirmed in Section 4. This was shown to be true not only with respect to the standard reconstruction, but also with respect to already-known TV-based reconstruction methods.

Moreover, a parallel implementation for multi-core CPUs and GPUs showed that this reconstruction process can be realized sufficiently fast in order to make the method also applicable in practice.

Motivated by these promising results, the focus of further research will be the development of a theory for a more general model using the TGV functional of arbitrary order.

ACKNOWLEDGEMENTS

REFERENCES

Alter, F., Durand, S., and Froment, J. (2005). Adapted total variation for artifact free decompression of JPEG images. *Journal of Mathematical Imaging and Vision*, 23:199–211.

Bredies, K. and Holler, M. (2011). A total-variation-based JPEG decompression model. Submitted for publication. Preprint available at http://math.uni-graz.at/mobis/publications.html.

Bredies, K., Kunisch, K., and Pock, T. (2010). Total generalized variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526.

Bredies, K. and Valkonen, T. (2011). Inverse problems with second-order total generalized variation constraints. In *Proceedings of SampTA*.

Caselles, V., Chambolle, A., and Novaga, M. (2007). The discontinuity set of solutions of the TV denoising problem and some extensions. *Multiscale Model. Simul.*, 6:879–894.

Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40:120–145.

Dawgbyte77 (2008). http://www.flickr.com/photos/dawg-byte77/3052164481/.

Frigo, M. and Johnson, S. G. (2005). The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231. Special issue on "Program Generation, Optimization, and Platform Adaptation".

Kartalov, T., Ivanovski, Z. A., Panovski, L., and Karam, L. J. (2007). An adaptive POCS algorithm for compression artifacts removal. In *9th International Symposium on Signal Processing and Its Applications*, pages 1–4.

Nikolova, M. (2000). Local strong homogeneity of a regularized estimator. *SIAM J. Appl. Math.*, 61:633–658.

Nosratinia, A. (2001). Enhancement of JPEG-compressed images by re-application of JPEG. *The Journal of VLSI Signal Processing*, 27:69–79.

NVIDIA (2008). NVIDIA CUDA programming guide 2.0. NVIDIA Cooperation.

OpenMP Architecture Review Board (2011). Openmp application program interface, version 3.1. http://www.openmp.org.

Ring, W. (2000). Structural properties of solutions to total variation regularization problems. *M2AN Math. Model. Numer. Anal.*, 34:799–810.

Shen, M.-Y. and Kuo, C.-C. J. (1998). Review of postprocessing techniques for compression artifact removal. *Journal of Visual Communication and Image Representation*, 9(1):2–14.

Singh, S., Kumar, V., and Verma, H. K. (2007). Reduction of blocking artifacts in JPEG compressed images. *Digital Signal Processing*, 17(1):225–243.

Wallace, G. K. (1991). The JPEG still picture compression standard. *Commun. ACM*, 34(4):30–44.

Weerasinghe, C., Liew, A. W.-C., and Yan, H. (2002). 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):891–897.

Zhong, S. (1997). Image coding with optimal reconstruction. In *International Conference on Image Processing*, volume 1, pages 161–164.

Zou, J. J. and Yan, H. (2005). A deblocking method for BDCT compressed images based on adaptive projections. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(3):430–435.