mathematics

MDPI

Article

Feature Keypoint-Based Image Compression Technique Using a Well-Posed Nonlinear Fourth-Order PDE-Based Model

Tudor Barbu

Institute of Computer Science of the Romanian Academy—Iași Branch, 700481 Iași, Romania;
tudor.barbu@iit.academiaromana-is.ro

Received: 30 April 2020; Accepted: 5 June 2020; Published: 7 June 2020

Abstract: A digital image compression framework based on nonlinear partial differential equations (PDEs) is proposed in this research article. First, a feature keypoint-based sparsification algorithm is proposed for the image coding stage. The interest keypoints corresponding to various scale-invariant image feature descriptors, such as SIFT, SURF, MSER, ORB, and BRIEF, are extracted, and the points from their neighborhoods are then used as sparse pixels and coded using a lossless encoding scheme. An effective nonlinear fourth-order PDE-based scattered data interpolation is proposed for solving the decompression task. A rigorous mathematical investigation of the considered PDE model is also performed, with the well-posedness of this model being demonstrated. It is then solved numerically by applying a consistent finite difference method-based numerical approximation algorithm that is next successfully applied in the image compression and decompression experiments, which are also discussed in this work.

Keywords: lossy image (de)compression; scale-invariant feature keypoint; image sparsification; scattered data point interpolation; nonlinear PDE-based structural inpainting; well-posed PDE model; variational solution; finite-difference-based numerical approximation

1. Introduction

The digital image compression represents an important image processing and analysis task whose purpose is to reduce the image file size without losing much information and while conserving its visual quality, so as to facilitate the image storing and transmission operations. The compression process, which could have an either lossless or lossy character, encodes the image content using fewer bits than its original representation [1].

The lossless compression algorithms remove or reduce considerably the statistical redundancy and recover exactly the image at decompression, without information loss. The following encoding algorithms are used for this type of compression: Huffman coding; aritmetic coding; run length encoding (RLE); LZW (Lempel–Ziv–Welch) encoding; or area coding, which is an enhanced 2D version of RLE [1–3]. Several well-known image formats, such as BMP, GIF, or PNG, are constructed using the lossless compression.

Lossy compression methods are characterized by higher compression rates, but they also lose an amount of image information. While the decompressed image versions are not identical to the originals, they look very similar. The lossy coding techniques include vector quantization, transformation-based encoding, fractal coding, block truncation coding, and sub-band coding [2,3]. Some very popular image standards based on lossy compression include JPEG, JPEG 2000, and other JPEG variants, which use transform-based coding (discrete cosine transform (DCT)- and discrete wavelet transform (DWT)-based image encoding) [1–3], and the MPEG standard with its versions [2,3].

The image compression techniques using partial differential equations (PDEs) represents a recently developed category of lossy compression methods. In the last years, PDEs have been increasingly applied in many static and video image processing and analysis fields, including image restoration [4], interpolation [5], segmentation [6], compression [7], registration [8], and optical flow estimation [9]. The PDE-based compression domain may also be considered as an application area of PDE-based inpainting (image reconstruction), because those compression algorithms use the PDE image interpolation schemes for decompression. The PDEs are rarely applied in the compression stage, usually being used by the pre-processing operations that prepare the image for the compression task. Thus, various coding techniques, such as random sparsification based coding, B-tree triangular coding (BTTC) [10], tree-based rectangular subdivision-based encoding [11], edge-based image coding [3,12], and clustering-based quantization [13], can be applied for image compression.

Depending on the inpainting solutions used in the decompression stage, various PDE-based compression techniques have been developed in the last decades. They include linear homogeneous diffusion-based approaches using the harmonic [12,14,15], biharmonic [15], and triharmonic inpainting operators [16], which provide successful results for cartoon compression [14], and depth map compression [12]. Moroever, some effective compression solutions have been obtained using the absolute monotone Lipschitz extension (AMLE) inpainting model [1,16] and the nonlinear anisotropic diffusion-based interpolation schemes derived from Perona-Malik [17], Charbonnier [18], total variation (TV) [19], and robust anisotropic diffusion (RAD) models [20].

More performant PDE-based image compression frameworks are those based on edge-enhancing diffusion (EED). Such a compression solution is the BTTC-EED image encoder introduced by Galic, Weickert et al. in 2005 [21]. It uses an edge-enhancing diffusion-based interpolation and an adaptive B-tree triangular coding-based image sparsification [10].The BTTC-EED compression method outperforms JPEG standard, when the two are compared at the same high compression rate, but it is outperformed by JPEG 2000 coder. An improved version of this encoder is Q64 + BTTC(L)-EED, proposed by Galic et al. in 2008 [22]. The most successful EED-based compression technique is the rectangular subdivision with edge-enhancing diffusion (R-EED) codec developed by Schmaltz et al. [11]. It clearly outperforms the BTTC-EED schemes and other PDE-based compression algorithms, as well as the JPEG codec. R-EED also outperforms JPEG 2000 for gray-level images, when compared at high compression ratios. An improved version of it, which was introduced by P. Pascal and J. Weickert in 2014 for color image compression [23], outperforms JPEG 2000 for color images as well, at high compression rates.

We conducted a high amount of research in the PDE-based image processing and analysis domains, and constructed numerous variational and differential models for image filtering [24], inpainting [25], segmentation [26], and compression [27]. A novel PDE-based image compression framework is considered here. A feature descriptor keypoint-based image sparsification, followed by a lossless RLE-inspired sparse pixel encoding process, is performed in the compression stage, which is described in the next section. While other compression techniques based on the scale-invariant descriptors, like SIFT, KAZE, and others, use their feature vectors [28–30], our novel method uses only the locations of their key points. As the number of these keypoints is not high, the proposed scheme has the advantage of producing high compression rates with still good decompression results.

The sparse pixel decoding and the inpainting of the decoded sparse image are performed in the decompression stage, which is described in the third section. The proposed nonlinear fourth-order PDE model for structure-based interpolation, representing the main contribution of this work, is presented in its second subsection. This inpainting algorithm provides the main advantage of the proposed method compared with many other PDE-based compression solutions, producing better interpolation on the same sparse image. A mathematical treatment concerning its well-posedness is performed on this model in the third subsection, with the existence of a unique variational solution being rigorously demonstrated. Another original contribution is the consistent finite difference-based numerical approximation scheme that converges to that weak solution is then described. Our image compression experiments and the comparisons with some well-known codecs are discussed in the fourth section, while the conclusions of this research are drawn in the final section of this article.

2. Interest Keypoint-Based Image Compression Algorithm

The compression component of the proposed framework consists of two main processes: image sparsification and sparse pixel coding. We could also add a pre-processing step that performs image enhancement using our past PDE-based filtering solutions [24]. The general scheme of the proposed compression algorithm is displayed in Figure 1.

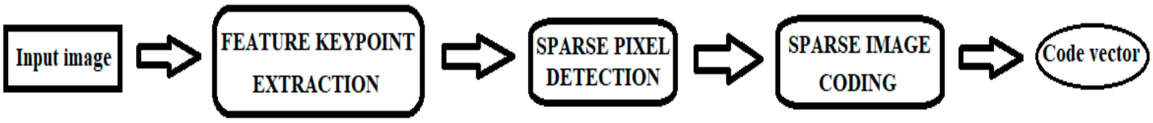


Figure 1. The scheme of the image compression process.

So, first we propose a sparsification algorithm based on the interest keypoints of the image, which represent the keypoints of the most important scale-invariant image feature descriptors. Thus, we are interested only in the locations of these key points, as they represent locally high information contents, not in the feature vectors corresponding to those descriptors that are successfully used in many computer vision tasks such as image/video object detection, recognition, and tracking. This idea of using only these locations for the sparsification task is a rather new one in the image compression field. So, we perform a keypoint detection process for each of the following feature descriptors: SIFT, SURF, MSER, BRISK, ORB, FAST, KAZE, and Harris.

Scale Invariant Feature Transform (SIFT), introduced by David Lowe in 1999 [31], represents the most renowned local feature description algorithm. The SIFT keypoints are extracted as the maxima and minima values of the result of difference of Gaussians (DoG) filters applied at different scales on the image. Next, a dominant orientation is then assigned to each detected keypoint and a local feature descriptor is computed for it. The detected features are invariant to scaling, orientation, affine transforms, and illumination changes.

Speeded up Robust Features (SURF) is a fast image feature descriptor proposed by H. Bay et al. in 2008 [32]. The SURF keypoint detection is based on a Hessian matrix approximation. Thus, the determinant of the Hessian matrix is used as a measure of local change around the point and the keypoints are chosen where this determinant is maximal. The orientations of these keypoints are then determined and square regions are extracted, centered on these interest points, and aligned to their orientations. The distribution-based SURF descriptor is then modelled by summing Haar wavelet responses extracted for [4 × 4] sub-regions of the interest square region.

Maximally Stable Extremal Regions (*MSER*) were introduced by Matas et al. in 2002 [33]. They are defined by an extremal property of the intensity function within the region and on its outer boundary and are used as a blob detection technique. Each MSER is represented by a keypoint, which is the position of a local intensity minimum (or maximum), and a threshold that is related to the intensity function.

Features from Accelerated Segment Test (*FAST*) represents a well-known feature detection algorithn developed by E. Rosten and T. Drummond [34]. It is successfully used for corner detection and is also used to construct other image feature descriptors, such as BRISK and ORB. *Binary Robust Invariant Scalable Keypoints* (*BRISK*) is an effective rotation and scale-invariant fast keypoint detection, description and matching algorithm [35]. The keypoint detection is performed by creating a scale space, computing FAST score across scale space, performing a pixel level non-maximal suppression, computing sub-pixel maximum across patch, and the continuous maximum across scales. Then, the image coordinates are reinterpolated from scale space feature point detection. Then, a rotation and scale invariant feature descriptor is computed for each detected keypoint. *Oriented FAST and rotated BRIEF* (*ORB*) represents another fast robust local feature detector, introduced by E. Rublee et al. in 2011 [36]. It is based on a combination of the FAST descriptor and the visual descriptor *BRIEF* (*Binary Robust Independent Elementary Features*) [37].

KAZE represents a multiscale 2D feature detection and description algorithm in nonlinear scale spaces [38]. The KAZE features are detected and described in a nonlinear scale space by using the nonlinear diffusion-based filtering. Harris Corner detector [39], introduced by C. Harris and M. Stephens in 1988 as an improvement of the Moravec's corner detector, is another well-known keypoint detection algorithm used by our sparsification approach.

A keypoint detection example is described in Figure 2.

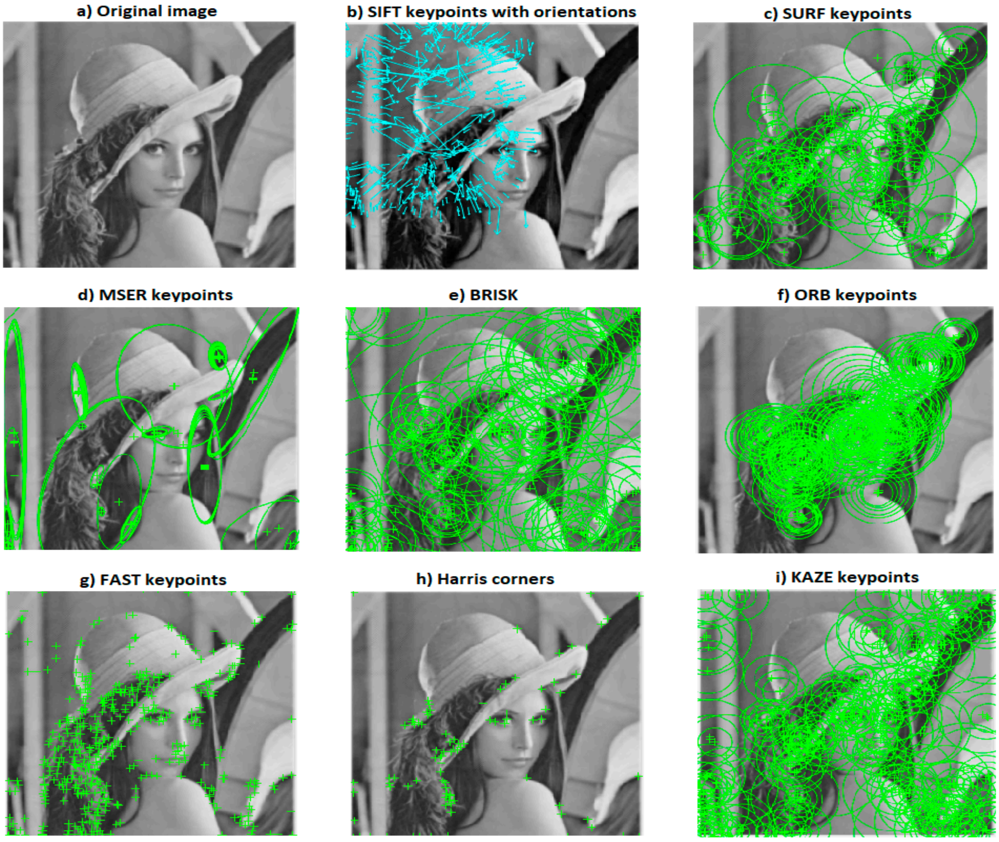


Figure 2. Feature keypoints corresponding to the most important descriptors. The feature keypoints of the original image in (**a**) are extracted, the keypoint detection results together with properties of their feature vectors being displayed in (**b**) SIFT, (**c**) SURF, (**d**) MSER, (**e**) BRISK, (**f**) ORB, (**g**) FAST, (**h**) Harris, and (**i**) KAZE.

Our compression approach extracts all the SIFT, SURF, MSER, BRISK, ORB, FAST, KAZE, and Harris keypoints of the analyzed image, but, depending on their number, may not use all of them in the coding process. Depending on their content information, some images could have a high number of keypoints, while other may have only few interest points. Using a high number of keypoints in the sparsification process would produce a very good decompression result, but also a low compression rate, while using a low number of keypoints would lead to a weaker decompression output, but to a higher compression rate.

So, as we are interested in an optimal trade-off between the compression rate and the decompressed image quality, if the number of the detected keypoints exceeds a properly selected threshold depending on the image size (for example, a third of all pixels), the proposed algorithm may consider only the strongest k keypoints of each descriptor for the sparsifcation. Otherwise, all the detected keypoints can be used as sparse pixels, unless their numbers become lower than another threshold depending on image size. In this case, one uses the neighborhoods of the keypoints to add more sparse pixels.

Thus, a pixel neighborhood based on a four- or an eight-connectivity can be considered around each keypoint. While using four-neighborhood provides higher compression ratios, the eight-neighborhoods lead to better decompression result. Larger square $[n \times n]$ neighborhoods having the pixels corresponding to keypoints as centers could also be used to obtain more spare pixels and achieve better decompression results, but they would lead to lower compression rates, too. When the pixel neighborhoods are used, the pixels representing interest keypoints are not taken into account, with only the pixels around them being used as sparse pixels. Moreover, in order to improve the decompression output, a minimum density of keypoints could be set. Thus, we may apply a grid to the image, and if the number of keypoints in the given $[c \times c]$ cell of this grid (with $c \in (10, 25)$) is lower than a certain threshold ((equal to 1, for example), then one or more keypoints should be randomly selected in that cell to be used as sparse pixels.

Then, the sparse image, which is identical to the original one in the sparse pixel's locations and black (only pixels with 0 value) elsewhere, is obtained. A lossless RLE-inspired encoding is then performed on this image sparsification result [2].

So, the image is converted into a 1D vector, V, using the raster scan order, then a code $C\ (V)$ is constructed for this vector. The proposed coding algorithm sets the current position in V at $i = 1$ and the code is initially a void vector $C\ (V)$: = [ ]. At each iteration, the coding procedure determines the 3-uple $\left[V(i), n_i, n_i^0\right]$, where

$$V(i) = V(i+1) = \ldots = V(i+n_i-1) \tag{1}$$

and

$$V(i+n_i) = \ldots = V\left(i+n_i+n_i^0-1\right) = 0 \tag{2}$$

The 3-uple is then appended to the code vector as follows:

$$C(V) := \left[C(V), V(i), n_i, n_i^0\right] \tag{3}$$

and the current position becomes $i := i + n_i + n_i^0$. When i becomes higher than the length of V, a final value, representing the ratio between the number of rows and the number of columns of the image, is added to $C\ (V)$. The final code stored in this vector $C\ (V)$ represents the image compression output. This lossless encoding algorithm may become a lossy coding scheme that improves the compression rates and ratios of (1) are replaced to $\|V(i+j) - V(i+j+1)\| \leq \textit{low threshold}, \forall j \in [0, n_i - 2]$.

3. Nonlinear PDE-Based Image Decompression Technique

Two main operations are performed by the decompression component of the developed framework: the decoding process, followed by the reconstruction one. The general scheme of the decompression algorithm is displayed in Figure 3.

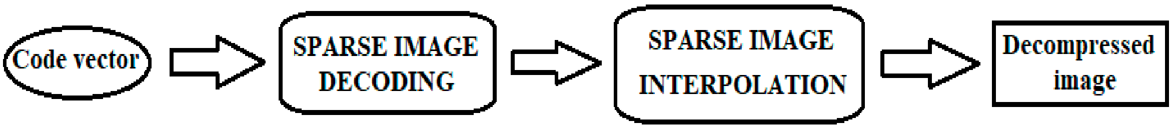


Figure 3. General scheme of the decompression framework.

3.1. Image Decoding Scheme

Thus, a decoding algorithm is applied to the values encoded and stored by $CV = C\ (V)$ computed by (3) to recover the vector V, which is initialized now as V: = [ ]. So, the code vector CV is processed iteratively and at each iteration i (starting with $i = 1$), a decoded sequence is added to the evolving vector as follows

$$V := \left[V, \left(\underbrace{CV(i), \ldots, CV(i),}_{CV\ (i+1)\ \text{times}} \ \underbrace{0...0}_{CV\ (i+2)\ \text{times}} \right) \right] \tag{4}$$

and the next current position in the code vector becomes $i := i + 3$.

This iterative decoding procedure continues while the value of i is lower than the length of CV. When i becomes equal to the length of the code vector CV, the obtained 1D vector V of length $l(V)$ is transformed into a $\left[l(V) \sqrt{\frac{l(V)}{CV(i)}} \times \sqrt{\frac{l(V)}{CV(i)}} \right]$ matrix, which represents the decoded sparse image.

Then, this result of the decoding process is further processed by the reconstruction procedure of the decompression scheme. In the reconstruction step, the initial image is recovered, as well as possible, from the scattered data points of the sparse image, by performing a PDE-based interpolation process, which is described in the next subsection.

3.2. Novel Fourth-Order Partial Differential Equation-Based Inpainting Model

We propose a nonlinear fourth-order PDE-based scattered data point interpolation technique that performs effectively the image reconstruction process [5,25]. It is based on the following PDE inpainting model with boundary conditions:

$$\begin{cases} \frac{\partial u}{\partial t} - \alpha\psi\left(\left|\nabla^2 u\right|\right)\nabla \cdot \left(\delta\left(\|\nabla^4 u_\sigma\|\right)\nabla u\right) + (1 - 1_\Gamma)(u - u_0) = 0, \text{on } (0,T) \times \partial\Omega \\ u(0,x,y) = u_0(x,y),\ \forall(x,y) \in \Omega \\ u(t,x,y) = 0,\ \text{on}(0,T) \times \partial\Omega \\ \Delta u(t,x,y) = 0,\ \text{on } (0,T) \times \partial\Omega \end{cases} \tag{5}$$

where $\alpha \in [1,2)$, $\Gamma \subset \Omega \subseteq R^2$ is the inpainting region that contains the black pixels of the sparse image represented here by the observed image $u_0 : \Omega \backslash \Gamma \to R$ and $u_\sigma = u * G_\sigma$, where $G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$ is the 2D Gaussian filter kernel [1].

We construct a diffusivity function for this model, which is positive, monotonic decreasing, and converging to 0, such as to achieve a proper diffusion process. It has the following form:

$$\delta : [0,\infty) \to [0,\infty), \delta(s) = \gamma \left(\frac{\eta(t,x,y)}{\left|\lambda \ln \eta(t,x,y) + \beta s^k\right|} \right)^{\frac{1}{k+1}} \tag{6}$$

where the coefficients are $\lambda, \gamma \in (0,1), \beta \in (1,3], k \geq 1$ and the conductance parameter $\eta(t,x,y)$ is a function depending on the coordinates and statistics of the evolving image and is computed by an algorithm that will be described later [24]. The Bi-Laplacian is used in the $\delta(\|\nabla^4 u_\sigma\|)$ component to overcome more effectively the staircase effect.

The other positive function used within this partial differential equation-based model has the following form:

$$\psi : [0,\infty) \to [0,\infty),\ \psi(s) = \frac{(\xi s^r + \nu)^{\frac{1}{r+1}}}{\zeta} \tag{7}$$

where its coefficients are $\xi, \nu, \zeta \in [1,3)$ and $r \in (0,1)$. The term $\psi\left(\left|\nabla^2 u\right|\right)$ is introduced to control the speed of the diffusion process and enhance the edges and other details.

The Dirichlet boundary conditions in (5) were chosen such that the proposed PDE model is well-posed. This fourth-order PDE-based interpolation scheme given by (5) reconstructs properly the observed image that is known only in a few locations, representing the sparse pixels. It performs the scattered pixel interpolation by directing the smoothing process toward the inpainting region and reducing it outside of that zone, using the inpainting mask given by the characteristic function $1_\Gamma(x,y)$ [5].

Thus, the reconstructed image would represent the solution of this nonlinear PDE model. So, the existence and unicity of such a solution is investigated in the following subsection, where the well-posedness of the proposed differential model is rigorously treated.

3.3. Mathematical Investigation of the Proposed PDE Model's Validity

In this subsection, one investigates the mathematical validity of the nonlinear fourth-order PDE-based model introduced in the previous subsection. Thus, a rigorous mathematical treatment is performed on the well-posedness of the proposed PDE inpainting model (5) to demonstrate the existence and unicity of a variational (weak) solution for it [4,40,41].

It is convenient to transform (5) into a parabolic equation of porous media type. So, we obtain the following PDE model:

$$\begin{cases} v_t - \alpha\Delta\psi(|v|)\nabla\cdot\left(\delta(\|G_1 * v\|)\cdot\nabla\left(\Delta^{-1}v\right)\right) + \Delta(1-1_\Gamma)\left(\Delta^{-1}v - u_0\right) = 0, \text{on } (0,T)\times\partial\Omega \\ v(0,x,y) = \Delta u_0(x,y),\ (x,y)\in\Omega \\ v(t,x,y) = 0,\ \text{on } (0,T)\times\partial\Omega \end{cases} \tag{8}$$

where $G_1 = \nabla^2 G_\sigma$ and $v = \nabla^2 u = \Delta u$.

We note that $\nabla\cdot\left(\delta(\|G_1 * v\|)\nabla\left(\Delta^{-1}v\right)\right) = \delta_s(\|G_1 * v\|)(\nabla G_1 * v)\cdot\nabla\left(\Delta^{-1}v\right) + \delta(\|G_1 * v\|)v + \nabla\cdot\delta(\|G_1 * v\|)$ and so (8) can be re-written as follows:

$$\begin{cases} v_t - \alpha\Delta\phi(x,y,v) + \Delta\left((1-1_\Gamma)\left(\Delta^{-1}v - u_0\right)\right) = 0, \text{on } (0,T)\times\partial\Omega \\ v(0,x,y) = v_0(x,y) = \Delta v_0(x,y),\ (x,y)\in\Omega \\ v(t,x,y) = 0,\ (x,y)\in\Omega, t\in(0,T) \end{cases} \tag{9}$$

where

$$\phi(x,y,v) = \psi(|v|)\delta_s(\|G_1 * v\|)(\nabla G_1 * v)\cdot\nabla\left(\Delta^{-1}v\right) + \alpha\delta(\|G_1 * v\|)v + \alpha\nabla\cdot\delta(\|G_1 * v\|) \tag{10}$$

Therefore, instead of (1), we shall study the existence problem for the nonlinear parabolic problem (9). The following hypotheses are easily verified [40,41]:

i. $\psi : [0,\infty) \to [0,\infty)$ is of class C^1 and $\beta_1 \le \psi'(r) \le \beta_2, \forall r \in R^+$ where $\beta_1, \beta_2 > 0$
ii. $\delta \in C^1(\Omega\times R), \delta > \rho > 0$, where ρ is a constant

By weak solution to problem (9), we mean a function $v : [0,T]\times\Omega \to R$, such that

$$v \in L^2\left(0,T;L^2(\Omega)\right)\cap C\left([0,T];H^{-1}(\Omega)\right), v_t \in L^2\left(0,T;\left(L^2\right)^*\right) \tag{11}$$

where $\left(L^2\right)^*$ is the dual space of $L^2(\Omega)$ with the pivotal space $H^{-1}(\Omega)$, which is the dual space of the Sobolev space $H_0^1(\Omega)$, and

$$\begin{aligned} &\int_\Omega v_t(t,x,y)(-\Delta)^{-1}\varphi(x,y)dxdy = \alpha\int_\Omega \phi(x,y,v(t,x,y))\varphi(x,y)dxdy \\ &-\int_\Omega (1-1_\Gamma)\left(\Delta^{-1}v(t,x,y) - u_0(x,y)\right)\varphi(x,y)dxdy = 0 \end{aligned} \tag{12}$$

for all $\varphi \in L^2(\Omega)$ [40,41].

We note, from (11) and (12), it follows that $\phi(\cdot,v) \in L^2\left(0,T;H_0^1(\Omega)\right)$, which implies $v = 0$ on $\partial\Omega$.

Theorem 1. *Assume that* $v_0 = \Delta u_0 \in L^2(\Omega)$ *and* $v_0 \geq 0$. *Then, under the hypotheses (i) and (ii), and for sufficiently small T, there exists at least one weak solution* $v \in C\Big([0,T]; H^{-1}(\Omega)\Big) \cap L^2\Big([0,T]; H_0^1(\Omega)\Big), v \geq 0$ *to Equation (9).*

Proof. We consider the following set:

$$\aleph = \left\{ v \in C\Big([0,T]; H^{-1}(\Omega)\Big); \|v(t)\|_{H^{-1}(\Omega)} \leq M, \int\limits_0^T \int\limits_\Omega \Big|v(t,x,y)\Big|^2 dtdxdy \leq M, v \geq 0 \; on(0,T) \times \Omega \right\}$$ For each $w \in \aleph$, consider the following equation:

$$\begin{cases} v_t - \alpha\Delta(\psi(v)\varsigma + \delta(\|G_1 * v\|)v + \nabla \cdot \delta(\|G_1 * v\|)) + \Delta(1 - 1_\Gamma)\Big(\Delta^{-1}v - u_0\Big) = 0 \\ v(0,x,y) = v_0(x,y) = \Delta u_0(x,y) \\ v(t,x,y) = 0 \text{ on } (0,T) \times \partial\Omega \end{cases} \tag{13}$$

where $\varsigma = \nabla \cdot \Big(\delta_s(\|G_1 * v\|)\nabla\Big(\Delta^{-1}w\Big)\Big)$. For each $w \in \aleph$, the operator $A(t) : L(\Omega) \to \Big(L^2\Big)^*$ defined by ${}_{(L^2)^*}(A(t)v, \varphi)_{L^2(\Omega)} = \alpha\int\limits_\Omega \psi(v)\varsigma\varphi dxdy - \int\limits_\Omega (1 - 1_\Gamma)\Big(\Delta^{-1}v - u_0\Big)\varphi dxdy + \alpha\int\limits_\Omega (\delta(\|G_1 * v\|)v + \nabla \cdot \delta(\|G_1 * v\|))\varphi dxdy$ for $\forall\varphi \in L^2(\Omega)$ is monotone, continuous, and coercive, that is, ${}_{(L^2)^*}(A(t)v, \varphi)_{L^2(\Omega)} = C_1\|\varphi\|^2_{L^2(\Omega)} - C_2\|\varphi\|^2_{(L^2)^*}$.

Then, by Theorem 1.2., Chapter 2 in [41], it follows that Equation (13), that is,

$$\begin{cases} \frac{\partial v}{\partial t} + A(t)v = 0, t \in (0,T) \\ v(0,x,y) = v_0(x,y) \end{cases} \tag{14}$$

has a unique solution $v = F(w)$, $v \in L^2\Big(0,T; L^2(\Omega)\Big)$, $\frac{\partial v}{\partial t} \in L^2(0,T; (L_2)^*)$. We are going to show that F has a fixed point $w = F(w)$. If multiply the Equation (13) by v and integrate it on $(0,t) \times \Omega$, we get the estimate for each $w \in \aleph$:

$${}_{(L^2)^*}(A(t)v, v)_{L^2(\Omega)} \geq \varphi\|v\|^2_{L^2(\Omega)} - C \tag{15}$$

where the constant C > 0 and, therefore, we get

$$\|v(t)\|^2_{H^{-1}(\Omega)} + s\int\limits_0^t \int\limits_\Omega \Big|v(s,x,y)\Big|^2 dsdxdy \leq C(M+1)T \tag{16}$$

Hence, for sufficiently small T and suitable M chosen, we have $F(w) \in \aleph, \forall w \in \aleph$. Moreover, F is continuous from $L^2\Big(0,T; H^{-1}(\Omega)\Big)$ to itself and, as easily seen, $F(\aleph)$ is compact in $L^2\Big(0,T; H^{-1}(\Omega)\Big)$, because by (13), we see that $\|\frac{dv}{dt}\|_{L^2(0,T;(L^2)^*)} \leq C, \forall w \in \aleph$. Then, by the classical Schauder's fixed point theorem [41], because $F(\aleph) \subset \aleph$, where $\aleph$ is a bounded convex set of $v \in L^2\Big(0,T; H^{-1}(\Omega)\Big)$ and $F(\aleph)$ is compact, there is $w* \in \aleph$ such that $F(w*) = w*$. Clearly, $u = w*$ is a solution to (8) as claimed. By Theorem 1, it follows a corresponding existence result for (5) via the transformation $u = \Delta^{-1}v$; therefore, the proposed PDE model is well-posed. □

3.4. Finite Difference-Based Numerical Approximation Scheme

The unique weak solution of the proposed PDE-based inpainting model, whose existence has been demonstrated in 3.3, is determined by solving numerically the model given by (5)–(7). So, a consistent numerical approximation scheme is developed for this well-posed PDE model by applying the finite difference method [42].

We consider a grid of space size h and time step Δt for this purpose. The space coordinates are quantized as $x = ih, y = jh, i \in \{1, \ldots, I\}, j \in \{1, \ldots, J\}$, while the time coordinate is quantized as $t = n\Delta t, n \in \{0, \ldots, N\}$, where the support image has the size of $[Ih \times Jh]$.

The nonlinear partial differential equation in (1) is discretized using finite differences [4,39]. It can be re-written as follows:

$$\frac{\partial u}{\partial t} + (1 - 1_\Gamma)(u - u_0) = \alpha\psi\left(\left|\nabla^2 u\right|\right)\nabla \cdot \left(\delta\left(\|\nabla^4 u_\sigma\|\right)\nabla u\right) \tag{17}$$

Its left term is then approximated numerically using forward difference for the time derivative [42]. We get the next discretization for it, $\frac{u_{i,j}^{n+\Delta t} - u_{i,j}^n}{\Delta t} + \lambda(1 - 1_\Gamma)\left(u_{i,j}^n - u_{i,j}^0\right)$. Then, the right term is approximated using the discrete Laplacian and the discretization of the divergence component. First, one computes

$$\psi_{i,j} = \psi\left(\|\Delta u_{i,j}^n\|\right) = \psi\left(\frac{u_{i+h,j}^n + u_{i-h,j}^n + u_{i,j+h}^n + u_{i,j-h}^n - 4u_{i,j}^n}{h^2}\right) \tag{18}$$

The divergence component can be re-written as

$$\nabla \cdot \left(\delta\left(\|\nabla^4 u_\sigma\|\right)\nabla u\right) = div\left(\delta\left(\left|\Delta(\Delta u_\sigma)\right|\right)\nabla u\right) = \frac{\partial}{\partial x}\left(\delta\left(\left|\Delta(\Delta u_\sigma)\right|\right)u_x\right) + \frac{\partial}{\partial y}\left(\delta\left(\left|\Delta(\Delta u_\sigma)\right|\right)u_y\right) \tag{19}$$

and is discretized as $\delta_{i+\frac{h}{2},j}\left(u_{i+h,j}^n - u_{i,j}^n\right) - \delta_{i-\frac{h}{2},j}\left(u_{i,j}^n - u_{i-h,j}^n\right) + \delta_{i,j+\frac{h}{2}}\left(u_{i,j+h}^n - u_{i,j}^n\right) - \delta_{i,j-\frac{h}{2}}\left(u_{i,j}^n - u_{i,j-h}^n\right)$, where

$$\delta_{i\pm\frac{h}{2},j} = \frac{\delta_{i\pm h,j} + \delta_{i,j}}{2}, \delta_{i,j\pm\frac{h}{2}} = \frac{\delta_{i,j\pm h} + \delta_{i,j}}{2} \tag{20}$$

$$\delta_{i,j} = \delta\left(\|\Delta(\Delta u_\sigma)_{i,j}^n\|\right) = \delta\left(\frac{\Delta(u_\sigma)_{i+h,j}^n + \Delta(u_\sigma)_{i-h,j}^n + \Delta(u_\sigma)_{i,j+h}^n + \Delta(u_\sigma)_{i,j-h}^n - 4\Delta(u_\sigma)_{i,j}^n}{h^2}\right) \tag{21}$$

where $\Delta(u_\sigma)_{i,j}^n = \frac{(u_\sigma)_{i+h,j}^n + (u_\sigma)_{i-h,j}^n + (u_\sigma)_{i,j+h}^n + (u_\sigma)_{i,j-h}^n - 4(u_\sigma)_{i,j}^n}{h^2}$. From (6), we have

$$\delta_{i,j} = \gamma\left(\frac{\eta(n,i,j)}{\left|\lambda \ln \eta(n,i,j) + \beta\|\Delta(\Delta u_\sigma)_{i,j}^n\|^k\right|}\right)^{\frac{1}{k+1}} \tag{22}$$

where the conductance parameter is determined as follows:

$$\delta(i,j,n) = \left|\mu\left(\|\nabla(u_\sigma)_{i,j}^{n-1}\|\right) + \kappa n\right| \tag{23}$$

where $\kappa \in (0,1]$.

Therefore, we get the following numerical approximation:

$$\begin{gathered} \frac{u_{i,j}^{n+\Delta t} - u_{i,j}^n}{\Delta t} + \lambda(1 - 1_\Gamma)\left(u_{i,j}^n - u_{i,j}^0\right) = \alpha\psi_{i,j} \\ \left[\delta_{i+\frac{h}{2},j}\left(u_{i+h,j}^n - u_{i,j}^n\right) - \delta_{i-\frac{h}{2},j}\left(u_{i,j}^n - u_{i-h,j}^n\right) + \delta_{i,j+\frac{h}{2}}\left(u_{i,j+h}^n - u_{i,j}^n\right) - \delta_{i,j-\frac{h}{2}}\left(u_{i,j}^n - u_{i,j-h}^n\right)\right] \end{gathered} \tag{24}$$

One may consider the parameter values $\Delta t = h = 1$. So, (24) leads to the next explicit iterative numerical approximation algorithm:

$$\begin{gathered} u_{i,j}^{n+1} = u_{i,j}^n(\lambda(1_\Gamma - 1) + 1) + \lambda u_{i,j}^0 + \\ \alpha\psi_{i,j}\left[\delta_{i+\frac{1}{2},j}\left(u_{i+1,j}^n - u_{i,j}^n\right) - \delta_{i-\frac{1}{2},j}\left(u_{i,j}^n - u_{i-1,j}^n\right) + \delta_{i,j+\frac{1}{2}}\left(u_{i,j+1}^n - u_{i,j}^n\right) - \delta_{i,j-\frac{1}{2}}\left(u_{i,j}^n - u_{i,j-1}^n\right)\right] \end{gathered} \tag{25}$$

The numerical approximation scheme (25) is applied to the evolving image u for n taking values from 0 to N. It is consistent to the nonlinear fourth-order PDE inpainting model (5) and converges to its variational solution representing the recovered image. The number of iterations required by a proper reconstruction is quite high (hundreds of steps), depending on the number of sparse pixels and the size of the observed image.

The final inpainting result, u^{N+1}, also represents the lossy image decompression output. The obtained decompressed image is not identical to the initial image, because of the loss of information produced by our PDE-based framework, but has a high visual similarity.

4. Experiments and Method Comparison

The nonlinear PDE-based compression framework proposed here was applied successfully on hundreds of digital images. Well-known image collections, such as the volumes of the USC-SIPI database and *the Image Repository of University of Waterloo* [43], were used in our simulations. These simulations were conducted on an Intel (R) Core (TM) i7-6700HQ CPU 2.60 GHz processor on 64 bits, operating Windows 10. The MATLAB environment was used for the software implementation of the described numerical interpolation algorithm.

It works properly for both clean and noisy images and achieves good values for the compression's performance measures, such as the compression rate (in bits per pixel bpp), the compression ratio, and the fidelity or quality. The described technique could achieve higher compression ratios by modifying some coding-related operations (see the discussions in the second section about using larger sparse pixel neighborhoods, transforming (1) and further processing the code vector), but the fidelity or quality measure may be negatively affected in those cases and decrease.

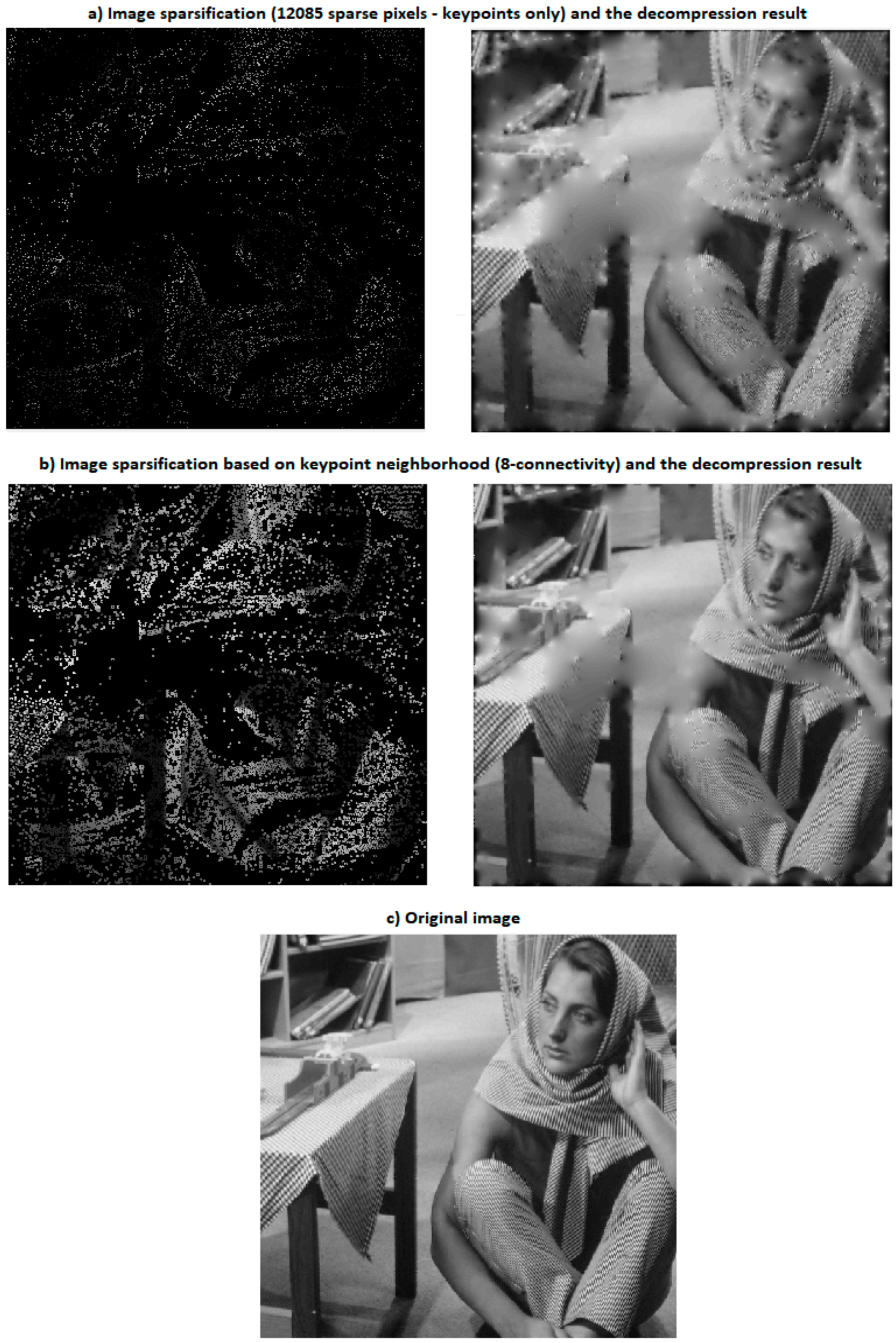
The compression component of this framework has a low computational complexity, as both the image sparsification and the sparse pixel encoding algorithm are characterized by low complexities and short execution times, of several seconds. The sparse pixel decoding approach also has a very low complexity, but the entire decompression technique is characterized by a much higher computational complexity, because of its more complicated nonlinear fourth-order PDE-based sparse pixel interpolation algorithm that performs a lot of computations, and a longer running time.

The performance of our decompression technique was assessed by applying similarity measures such as the peak signal-to-noise ratio (PSNR), mean squared error (MSE), and structural similarity index (SSIM) [44]. The decompressed versions of the images encoded using the proposed approach achieve good values of these metrics when measured against their originals. The compression rates obtained from the performed experiments lie between 0.30 and 0.75 bpp, which corresponds to compression ratios from 20:1 down to 11:1.

Some examples of image compression processes performed using our framework are described in Figure 4.

Compression method comparison was also performed. We compared the proposed approach to other image compression techniques, some of them based on PDEs, by applying similarity measures [44], like PSNR, MSE, and SSIM, for them at the same compression rates.

The compression component of our framework, using a scale-invariant feature keypoint-based image sparsification and an effective sparse pixel encoding algorithm, outperforms the compression schemes using other scattered data point selection and coding methods, such as the random sparsification, the edge-based sparsification and coding, or even the BTTC, providing better compression rates. The proposed nonlinear fourth-order PDE-based decompression technique provides better results than other PDE inpainting models, achieving better PSNR and SSIM values, although it has some disadvantages of its own. Given the high complexity of the interpolation algorithm (25), which requires many processing steps, our decompression scheme does not execute very fast. It may need more than 1 min for large images. Moreover, given the structural character of the PDE-based inpainting model (5), the proposed framework does not properly decompress the image textures.



a) Image sparsification (12085 sparse pixels - keypoints only) and the decompression result

b) Image sparsification based on keypoint neighborhood (8-connectivity) and the decompression result

c) Original image

Figure 4. Image (de)/compression examples based on the proposed technique. The original [512 × 512] *Barbara* image displayed in (**c**) is encoded using the sparsification in the first image of (**a**), based on the pixels corresponding to the 12,085 feature keypoints only and producing a code vector of 35,203 coefficients and a compression ratio of 13:1, with the decompression result being displayed in the second image of (**a**). The compression process described in (**b**) produces a better decompression output, as it uses an image sparsification based on the eight neighborhoods of the feature keypoints, which means 54,364 scattered interpolation points, a code vector with 121,021 values, and a lower compression ratio of 2.2:1.

Thus, we compared our PDE-based interpolation algorithm to other well-known PDE inpainting models of various orders in the decompression stage, with the same sparsification solution being chosen for compression. So, image interpolation schemes based on biharmonic smoothing, TV inpainting, and the fourth-order You–Kaveh model [45] adapted for inpainting were considered. Keypoint-based, edge-based, and random sparsification solutions were used in the coding stage. The PDE inpainting model proposed here outperforms the other PDE-based schemes, providing better values of the performance metrics (average PSNR) in all these sparsification cases, as it results from Table 1. It also results from the following table that our feature keypoint-based sparsification technique outperforms both the random keypoint selection and the edge-based sparsification schemes.

Table 1. Method comparison: the performance of several partial differential equation (PDE) inpainting models in decompression stage. TV, total variation.

Sparsification Technique	The Proposed PDE Scheme	Biharmonic Inpainting	TV Inpainting	Fourth-Order PDE Model
Keypoint-based sparsification (described here)	25.3475 (dB)	24.4917 (dB)	22.9432 (dB)	23.4215 (dB)
Edge-based sparsification	25.1853 (dB)	24.3421 (dB)	22.1975 (dB)	22.1484 (dB)
Random sparsification (8% pixels)	20.2316 (dB)	18.9323 (dB)	16.9824 (dB)	17.1544 (dB)

The obtained values of the similarity metrics illustrate the effectiveness of the nonlinear PDE-based compression method described here that outperforms some well-known compression algorithms. Thus, our framework provides better decompression results than image compression techniques using random sparsification, edge-based coding, or B-tree triangular coding with PDE inpainting models like those using harmonic, biharmonic, and triharmonic smoothing, AMLE inpainting, and the second-order anisotropic diffusion-based inpainting schemes. It also outperforms the BTTC-L compression [10] and performs slightly better than the BTTC-EED model and the JPEG standard, at higher compression rates, while it is outperformed by the JPEG 2000 codec and the R-EED compression.

Several other method comparison results are displayed in Figure 5 and Table 2. Thus, the next figure contains the results obtained by some compression approaches on the [256 × 256] Trui image, at a compression rate of 0.40 bits/pixel.

Table 2. Average peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) values obtained by various methods at 0.4 bpp compression rate. BTCC, B-tree triangular coding; EED, edge-enhancing diffusion.

Compression Method	Average PSNR	Average SSIM
The proposed fourth-order PDE-based technique	26.3475 (dB)	0.6815
Harmonic inpainting-based compression	20.8495 (dB)	0.5167
Charbonnier diffusion-based model	22.3321 (dB)	0.5534
BTTC-L	23.1485 (dB)	0.6102
BTTC-EED	26.2378 (dB)	0.6723
JPEG	26.1075 (dB)	0.6546
JPEG 2000	26.8379 (dB)	0.7047
R-EED	27.1465 (dB)	0.7161

The decompression output of the nonlinear PDE-based method proposed here is displayed in (i). The average PSNR and SSIM values obtained by the respective encoding approaches are displayed in Table 2. One can see that only JPEG 2000 and R-EED produce higher, but still comparable, metric values than the described compression technique.

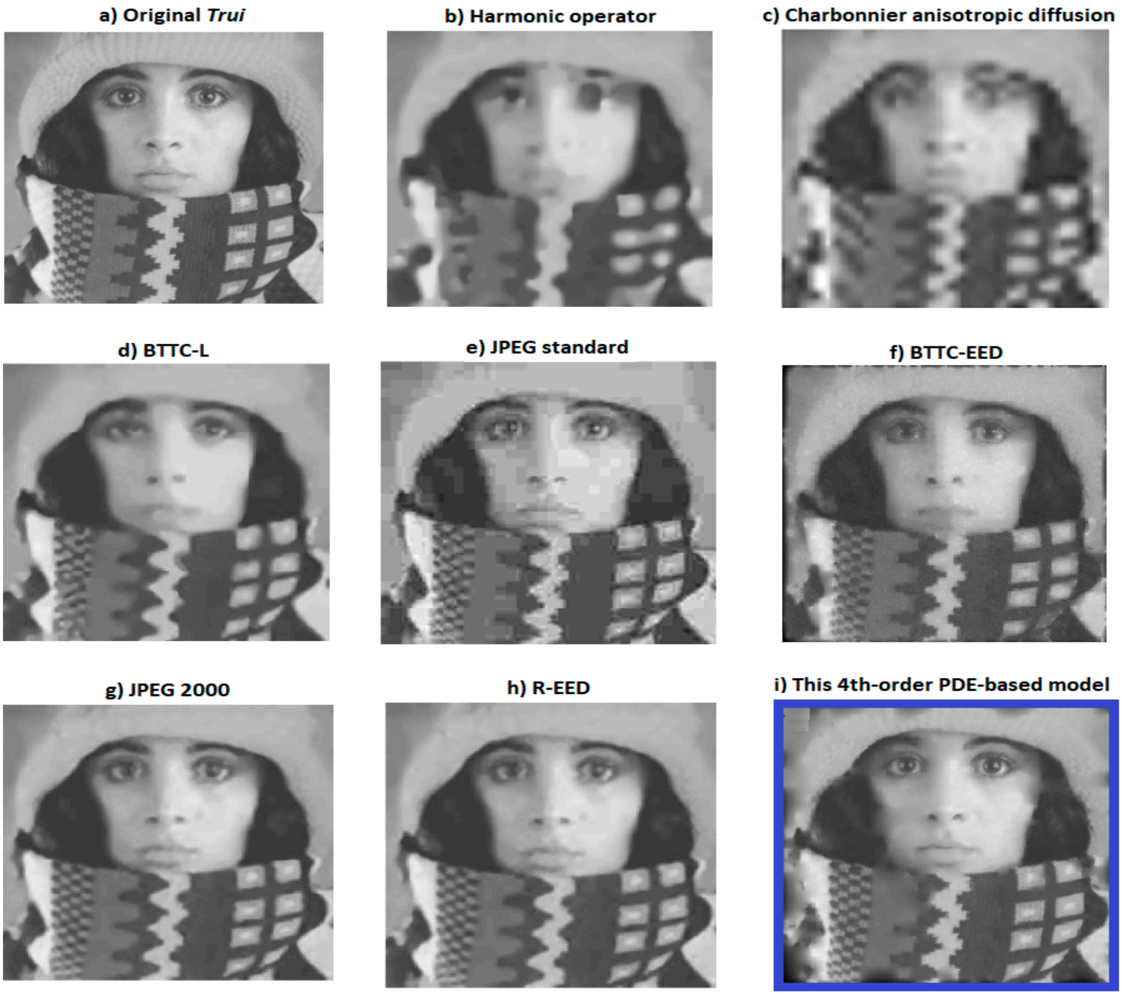


Figure 5. Image decompression results achieved by several techniques at a compression rate of 0.4 bpp: (**a**) Original image, (**b**) Harmonic Inpainting-based compression, (**c**) Charbonnier diffusion-based compression, (**d**) BTTC-L, (**e**) JPEG, (**f**) BTTC-EED, (**g**) JPEG 2000, (**h**) R-EED, (**i**) Proposed approach.

5. Conclusions

A novel lossy image coding technique, which is based on the scale-invariant feature keypoint locations in the compression stage and a nonlinear fourth-order PDE-based model in the decompression stage, was proposed in this paper. The nonlinear partial differential equation-based model that was introduced, mathematically investigated, and numerically solved represents the major contribution of this work. In fact, it was the main purpose of this work to develop and analyze a new well-posed nonlinear partial differential equation-based interpolation model that can be successfully applied in the image compression domain. Beyond compression, however, the proposed PDE scheme can be successfully used for solving any inpainting task, like object or text removal, and an adapted version of it can be used efficiently in the image denoising and restoration domain.

While the PDE-based image decompression schemes are usually based on second-order diffusion equations, we proposed a more complex nonlinear fourth-order PDE inpainting model for the decompression task. Given the 2D Gaussian filter kernel involved in its componence, it performs efficiently in both clean and noisy conditions. Its well-posedness was rigorously demonstrated here and a consistent finite difference method-based approximation scheme was constructed to determine numerically the unique variational solution of it.

While the most important and original contributions of the developed framework are related to its decompression component, some novelties are brought in the compression stage, too. Thus, we proposed a novel image sparsification solution that is based only on the locations of the SIFT, SURF, MSER, BRISK, ORB, FAST, KAZE, and Harris keypoints and not on their features, and a sparse pixel encoding algorithm that is inspired by RLE. Although the proposed PDE-based inpainting algorithm may provide better decompression results when used in combination with some other sparsification models, this new keypoint-based image sparsification method has the advantage of using a rather low number of scattered data points and still leading to a good output.

The performed experiments and method comparisons illustrate the effectiveness of the proposed technique. It outperforms many existing image compression models and provides comparable good results with some state-of-the-art approaches. It has some drawbacks related to the running time, which can be quite long for large images, given the high computational complexity of the keypoint detection and sparse pixel encoding, decoding, and interpolation algorithms, and to the structural character of the PDE-based inpainting scheme, which also leads to weaker decompression results for the textured images. So, investigating some solutions to reduce the complexity of this technique, especially in the decompression stage, to get a lower execution time, and to improve the texture compression results, will represent the focus of our future research in this field.

Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Gonzales, R.; Woods, R. *Digital Image Processing*, 2nd ed.; Prentice Hall: New York, NY, USA, 2001.
2. Sayood, K. *Introduction to Data Compression*, 3rd ed.; Morgan Kaufmann: Burlington, MA, USA, 2005.
3. Bhaskaran, V.; Konstantinides, K. *Image and Video Compression Standards*; Kluwer Academic Press: Boston, MA, USA, 1995.
4. Aubert, G.; Kornprobs, P. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*; Springer Science & Business Media: Berlin, Germany, 2006; Volume 147.
5. Schonlieb, C.B. *Partial Differential Equation Methods for Image Inpainting*; Cambridge University Press: Cambridge, UK, 2015; Volume 29.
6. Mumford, D.; Shah, J. Optimal approximation by Piecewise Smooth Functions and Associated Variational Problems. *Commun. Pure Appl. Math.* **1989**, *42*, 577–685. [CrossRef]
7. Peter, P.T. Understanding and Advancing PDE-Based Image Compression. Ph.D. Thesis, Saarland University, Saarbrücken, Germany, 2016.
8. Zhang, J.; Chen, K. Variational image registration by a total fractional-order variation model. *J. Comput. Phys.* **2015**, *293*, 442–461. [CrossRef]
9. Lee, S.; Park, E. A content adaptive fast PDE algorithm for motion estimation based on matching error prediction. *J. Commun. Netw.* **2010**, *12*, 5–10. [CrossRef]
10. Distasi, R.; Nappi, M.; Vitulano, S. Image compression by B-tree triangular coding. *IEEE Trans. Commun.* **1997**, *45*, 1095–1100. [CrossRef]
11. Schmaltz, C.; Weickert, J.; Bruhn, A. Beating the quality of JPEG 2000 with anisotropic diffusion. In *Pattern Recognition*; Lecture Notes in Computer Science; Springer: Berlin, Germany, 2009; Volume 5748, pp. 452–461.
12. Chen, J.; Ye, F.; Di, J.; Liu, C.; Men, A. Depth map compression via edge-based inpainting. In Proceedings of the 29th Picture Coding Symposium, Krakow, Poland, 9 May 2012; pp. 57–60.
13. Hoeltgen, L.; Peter, P.; Breuß, M. Clustering-based quantisation for PDE-based image compression. *Signal Image Video Process.* **2018**, *12*, 411–419. [CrossRef]
14. Mainberger, M.; Weickert, J. Edge-based image compression with homogeneous diffusion. In *Computer Analysis of Images and Patterns, Lecture Notes in Computer Science*; Springer: Berlin, Germany, 2009; Volume 5702, pp. 476–483.
15. Hoffmann, S.; Plonka, G.; Weickert, J. Discrete greens functions for harmonic and biharmonic inpainting with sparse atoms. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*; Springer: Cham, Switzerland, 2015; pp. 169–182.

16. Schmaltz, C.; Peter, P.; Mainberger, M.; Ebel, F.; Weickert, J.; Bruhn, A. Understanding, optimising, and extending data compression with anisotropic diffusion. *Int. J. Comput. Vis.* **2014**, *108*, 222–240. [CrossRef]
17. Perona, P.; Malik, J. Scale-space and edge detection using anisotropic diffusion. *Proc. IEEE Comput. Soc. Workshop Comput. Vis.* **1987**, *12*, 16–22. [CrossRef]
18. Charbonnier, P.; Blanc-Feraud, L.; Aubert, G.; Barlaud, M. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceeding IEEE International Conference on Image Processing*; IEEE Company Society Press: Austin, TX, USA, 1994; Volume 2, pp. 168–172.
19. Chan, T.F.; Shen, J. *UCLA CAM Report Morphologically Invariant PDE Inpaintings*; University of Minnesota Digital Conservancy: Minneapolis, MN, USA, 2001; pp. 1–15.
20. Black, M.; Shapiro, G.; Marimont, D.; Heeger, D. Robust anisotropic diffusion. *IEEE Trans. Image Process.* **1998**, *7*, 421–432. [CrossRef] [PubMed]
21. Galic, I.; Weickert, L.; Welk, M.; Bruhn, A.; Belyaev, A.; Seidel, H.-P. Image compression with anisotropic diffusion. *J. Math. Imaging Vis.* **2008**, *31*, 255–269. [CrossRef]
22. Galic, I.; Weickert, L.; Welk, M.; Bruhn, A.; Belyaev, A.; Seidel, H.-P. Towards PDE-based image compression. Variational, Geometric and Level-Set Methods in Computer Vision. In *Lecture Notes in Computer Science*; Springer: Berlin, Germany, 2005; Volume 3752, pp. 37–48.
23. Peter, P.; Weickert, J. Colour image compression with anisotropic diffusion. In Proceedings of the 21st IEEE International Conference on Image Processing, Paris, France, 14 February 2014; pp. 4822–4826.
24. Barbu, T.; Favini, A. Rigorous mathematical investigation of a nonlinear anisotropic diffusion-based image restoration model. *Electron. J. Differ. Equ.* **2014**, *129*, 1–9.
25. Barbu, T. Variational image inpainting technique based on nonlinear second order diffusions. *Comput. Electr. Eng.* **2016**, *54*, 345–353. [CrossRef]
26. Barbu, T. Robust contour tracking model using a variational level-set algorithm. *Numer. Funct. Anal. Optim.* **2014**, *35*, 263–274. [CrossRef]
27. Barbu, T. Segmentation-based non-texture image compression framework using anisotropic diffusion models. *Proc. Rom. Acad. Ser. A Math. Phys. Tech. Sci. Inf. Sci.* **2019**, *20*, 122–130.
28. Yue, H.; Sun, X.; Yang, J.; Wu, F. SIFT-based image compression. In Proceedings of the IEEE International Conference Multimedia Expo Workshops, Melbourne, Australia, 9–13 July 2012; pp. 473–478.
29. Srivastava, S.; Mukherjee, P.; Lall, B. Adaptive image compression using saliency and KAZE features. In Proceedings of the International Conference on Signal Processing and Communications, Bangalore, India, 25 January 2016.
30. Dosovitskiy, A.; Brox, T. Inverting visual representations with convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 6 June–1 July 2016.
31. Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 2, pp. 1150–1157.
32. Bay, H.; Ess, A.; Tuytelaars, T.; Gool, L.V. Speeded Up Robust Features. *Computer Vis. Image Underst. (CVIU)* **2008**, *110*, 346–359. [CrossRef]
33. Matas, J.; Chum, O.; Urban, M.; Pajdla, T. Robust wide baseline stereo from maximally stable extremal regions. *Image Vis. Comput.* **2004**, *22*, 761–767. [CrossRef]
34. Rosten, E.; Drummond, T. Machine learning for highspeed corner detection. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 1.
35. Leutenegger, S.; Chli, M.; Siegwart, R. BRISK: Binary robust invariant scalable keypoints. In Proceedings of the IEEE International Conference Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2548–2555.
36. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011.
37. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. Brief: Binary robust independent elementary features. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2010.
38. Alcantarilla, P.F.; Bartoli, A.; Davison, A.J. KAZE features. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 214–227.
39. Harris, C.; Stephens, M. A Combined Corner and Edge Detector. In Proceedings of the Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988; pp. 147–152.
40. Lions, L.; Strauss, W. Some non-linear evolution equations. In *Bulletin de la Société Mathématique de France*; Bulletin de la S.M.F.: Paris, France, 1965; Volume 93, pp. 43–96.

41. Lions, L. *Quelques Méthodes de Résolution des Problèmes aux Limites non Linéaires*; Dunod & Gauthier-Villars: Paris, France, 1969.
42. Johnson, P. *Finite Difference for PDEs*; Semester I; School of Mathematics, University of Manchester: Manchester, UK, 2008.
43. University of Waterloo, Fractal Coding and Analysis Group. Image Repository. Digital Image Collection. Available online: http://links.uwaterloo.ca/Repository.html (accessed on 23 December 2019).
44. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]
45. You, Y.L.; Kaveh, M. Fourth-order partial differential equations for noise removal. *IEEE Trans. Image Process.* **2000**, *9*, 1723–1730. [CrossRef] [PubMed]