THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 20, Number 2/2019, pp. 123–131

SEGMENTATION-BASED NON-TEXTURE IMAGE COMPRESSION FRAMEWORK USING ANISOTROPIC DIFFUSION MODELS

Tudor BARBU

Institute of Computer Science of the Romanian Academy
E-mail: tudor.barbu@iit.academiaromana-is.ro

Abstract. A novel partial differential equation (PDE) – based non-texture image compression and decompression technique is proposed in this work. The anisotropic diffusion schemes are used in both the compression and decompression stage of this framework. A nonlinear second-order diffusion-based edge detection is performed first. The pixels adjacent to the detected level lines are then extracted and encoded using a lossless coding procedure. The decompression task is performed successfully applying a combined nonlinear fourth-order PDE-based structural inpainting algorithm that recovers properly the image by interpolating the encoded key pixels. Consistent finite difference-based numerical approximation schemes are developed for the diffusion-based models used by the proposed compression approach. Some experiments and method comparisons are also discussed.

Key words: image de/compression, edge-based segmentation, anisotropic diffusion model, key pixel de/coding, structural PDE-based interpolation, finite difference-based numerical approximation.

1. INTRODUCTION

Image compression represents an important branch of both the image processing and the data compression domains. Its purpose is to reduce the file size of the image without losing too much information and maintaining a good level of visual quality, so that to facilitate its storage and transmission. The compression process involves encoding the image information by using fewer bits than the original representation and could be either lossy or lossless [1]. In the lossless compression techniques, which eliminate or reduce the statistical redundancy, the compressed image is recovered perfectly in the decompression stage, no information being lost. The next algorithms are included in the lossless compression: Huffman encoding, Run Length Encoding (RLE), Aritmetic coding, LZW encoding and Area coding [1-3]. Digital image formats such as GIF, BMP and PNG are based on lossless coding. The lossy coding methods provide much higher compression ratios than lossless techniques but the decompressed images produced by them are not identical to the original versions, but very closed. The lossy compression schemes include: Vector quantization, Transformation coding, Fractal coding, Block Truncation Coding and Sub-band coding [2,3]. Some popular image formats based on lossy compression include JPEG, JPEG 2008 and other JPEG versions [1-3], and MPEG with its versions (for movies) [2, 3].

A recently developed category of lossy image compression techniques is based on partial differential equations. In the last three decades, the PDE-based and variational models have been applied successfully in some important image processing and computer vision domains, such as image denoising and restoration [4], inpainting [5], segmentation [6], registration [7] and compression [8]. In fact, the PDE-based image compression field represents an obvious application area of the PDE-based inpainting, since the compression techniques use PDE interpolation schemes in the decompression stage. An influential PDE-based compression approach is that introduced by Galic et al. in 2005 [9]. It uses an edge-enhancing diffusion (EED) - based interpolation and an adaptive sparsification of the image data inspired by the triangulation from the B-tree triangular coding (BTTC) [10]. The most successful EED-based inpainting model is the rectangular subdivision with edge-enhancing diffusion (R-EED) codec proposed by Schmaltz et al. [11].

Other PDE-based image coding techniques, which use linear homogeneous diffusion models, provide successful results for cartoon compression [12] and depth map compression [13].

We have conducted a high amount of research in the PDE-based image processing and analysis domains, developing many techniques for restoration [14], inpainting [15] and segmentation [16]. Here we consider the PDE-based image compression domain, developing a novel diffusion-based technique. Unlike other PDE-based coding approaches, the proposed method uses nonlinear anisotropic diffusion models in both the compression and the decompression stage. Thus, the encoding module of the proposed framework, which is described in the next section, uses the edge information of the image, employing a second-order anisotropic diffusion-based scheme to extract the high-contrast level lines. The pixel information from the vicinity of these level lines is then coded and stored by using a lossless algorithm that achieves a good compression rate. The decompression module described in the third section uses a nonlinear PDE-based inpainting model that is combining second and fourth order diffusions. The proposed anisotropic diffusion-based structural interpolation technique reconstructs properly the non-texture image from the decoded set of pixels, but with some loss of information. Some successful image compression experiments are described in the fourth section, where method comparison are also discussed. Conclusions are drawn in the final section.

2. ANISOTROPIC DIFFUSION-BASED IMAGE CODING USING EDGE-BASED DATA

We consider an edge-based coding algorithm for the image compression component of this framework. Level lines characterized by a high contrast, the edges play an important role in image processing and analysis [1]. They contain meaningful information that can be exploited successfully for image compression. The proposed compression method extracts and encodes this semantically important edge-based information.

Thus, an edge detection process is performed first. Here, one may apply a classical edge detector, such as Sobel [1], Prewitt [1], Roberts [1], Canny [17] or the Marr-Hildreth operator [18]. Instead, we choose to construct a new edge-based image segmentation technique using a nonlinear second-order anisotropic diffusion-based edge-preserving smoothing scheme that is proposed here. The considered PDE-based model combines a parabolic second-order differential equation with boundary conditions to a 2D filter kernel:

$$\begin{cases} \dfrac{\partial u}{\partial t} - \delta\left(\left\|\nabla(K_{x,y} * u)\right\|\right)\nabla\cdot\left(\psi\left(\left\|\nabla u\right\|\right)\nabla u\right) + \alpha(u-u_0) = 0 \\ u(x,y,0) = u_0(x,y), \quad \forall (x,y) \in \Omega \\ u(t,x,y) = 0, \quad \forall (x,y) \in \partial\Omega \end{cases} \tag{1}$$

where $\alpha \in [0,1)$, the domain $\Omega \subseteq R^2$, u is the evolving image and the observed image $u_0 = \mathrm{Im} * K_{x,y}$, Im being the original image to be compressed. Convolving initial image to a 2D filter add an amount of blurring to it [1], facilitating the essential edge extraction. The edge-stopping function of the model is constructed as:

$$\psi : [0,\infty) \to [0,\infty), \quad \psi(s) = \varepsilon\left(\frac{\xi(u)}{\left|\beta s^2 + \zeta \ln \xi(u)\right|}\right)^{\frac{1}{3}} \tag{2}$$

where $\beta \in (2,4]$, ε, $\zeta \in (0,1)$ and the conductance parameter is modeled as following:

$$\xi(u) = \nu\left|\mu\left(\left\|\nabla u\right\|\right) + \mathrm{median}\left(\left\|\nabla u\right\|\right)\right| \tag{3}$$

with $\nu \in [1,3)$, $\mu(\cdot)$ represents the average and $\mathrm{median}(\cdot)$ returns the median value of the argument.

The edge-stopping function (2) is properly constructed for an edge-preserving denoising process, being positive, monotonically decreasing and convergent to zero: $\lim_{s\to\infty}\psi(s) = 0$ [4, 14]. The term $\delta\left(\left\|\nabla\left(K_{x,y} * u\right)\right\|\right)$ has the role of controlling the diffusion speed and enhancing edges. It uses a convolution between the evolving image and the two-dimension filter kernel and is based on the function:

$$\delta:[0,\infty)\to[0,\infty),\ \ \delta(s)=\kappa\sqrt[r+1]{\upsilon s^r+\rho} \tag{4}$$

with $\kappa,r\in(0,1)$, $\nu,\rho\in[1,2)$. A good filter kernel solution is 2D Gaussian filter $K_{x,y}=\frac{1}{2\pi\sigma^2}\mathrm{e}^{-\frac{x^2+y^2}{2\sigma^2}}$ [1].

This nonlinear anisotropic diffusion model provides an effective detail-preserving additive Gaussian noise removal. Given its strong edge-preserving character, this PDE-based model defines successfully the image edges and can be used to extract them properly. It is well-posed, admitting a unique weak solution that is computed using a finite difference-based numerical approximation algorithm constructed for it (1) [19]. We use a grid of space size h and time step size Δt to quantifies the time and space coordinates as:

$$x=i\,h,\ \ y=jt,\ \ t=n\,\Delta t,\ \ \ i\in\{1,...,I\},\ \ j\in\{1,...,J\},\ \ n\in\{0,...,N\}. \tag{5}$$

Since the nonlinear partial differential equation from (1) could be re-written in the form $\frac{\partial u}{\partial t}+\alpha(u-u_0)=\delta\left(\left\|\nabla\left(K_{x,y}*u\right)\right\|\right)\left(\frac{\partial}{\partial x}\left(\psi\left(\left|\nabla u\right|\right)u_x\right)+\frac{\partial}{\partial y}\left(\psi\left(\left|\nabla u\right|\right)u_y\right)\right)$, the left term is discretized using finite differences as $\frac{u_{i,j}^{n+\Delta t}-u_{i,j}^n}{\Delta t}+\alpha\left(u_{i,j}^n-u_{i,j}^0\right)=u_{i,j}^{n+\Delta t}\frac{1}{\Delta t}+u_{i,j}^n\left(\alpha-\frac{1}{\Delta t}\right)-u_{i,j}^0\alpha$. Then, the right term is approximated by using the central differences [19]. First, one computes $\delta_{i,j}=\delta\left(\left\|\left(K*u\right)_{i,j}\right\|\right)$, $\psi_{i,j}=\psi\left(\left\|u_{i,j}\right\|\right)$, where $\left\|u_{i,j}\right\|\approx\sqrt{\left(\frac{u_{i+h,j}-u_{i-h,j}}{2h}\right)^2+\left(\frac{u_{i,j+h}-u_{i,j-h}}{2h}\right)^2}$. Next, $\frac{\partial}{\partial x}\left(\psi\left(\left|\nabla u\right|\right)u_x\right)$ and $\frac{\partial}{\partial y}\left(\psi\left(\left|\nabla u\right|\right)u_y\right)$ are discretized spatially as $\psi_{i+\frac{h}{2},j}\left(u_{i+h,j}-u_{i,j}\right)-\psi_{i-\frac{h}{2},j}\left(u_{i,j}-u_{i-h,j}\right)$ and $\psi_{i,j+\frac{h}{2}}\left(u_{i,j+h}-u_{i,j}\right)-\psi_{i,j-\frac{h}{2}}\left(u_{i,j}-u_{i,j-h}\right)$, where $\psi_{i\pm\frac{h}{2},j}=\frac{\psi_{i\pm h,j}+\psi_{i,j}}{2}$, $\psi_{i,j\pm\frac{h}{2}}=\frac{\psi_{i,j\pm h}+\psi_{i,j}}{2}$. If $h=\Delta t=1$ we get the explicit iterative numerical approximation algorithm

$$\begin{aligned}u_{i,j}^{n+1}=&u_{i,j}^n(1-\alpha)+u_{i,j}^0\alpha+\delta_{i,j}\left(\psi_{i+\frac{1}{2},j}\left(u_{i+1,j}^n-u_{i,j}^n\right)-\psi_{i-\frac{h}{2},j}\left(u_{i,j}^n-u_{i-1,j}^n\right)+\right.\\&\left.+\psi_{i,j+\frac{1}{2}}\left(u_{i,j+1}^n-u_{i,j}^n\right)-\psi_{i,j-\frac{1}{2}}\left(u_{i,j}^n-u_{i,j-1}^n\right)\right)\end{aligned} \tag{6}$$

The iterative discretization scheme (6) is consistent to the anisotropic diffusion model (1), converging to its solution representing the denoised image. It is then applied for boundary detection, computing the state of the evolving image at two different iterations, let them be m and $m+T$, with T large enough, for example $T>30$. One may use a low value for m, for example $m=0$. Next, one computes the absolute difference between the two states. Then a thresholding procedure is applied to this difference image, by using the conductance parameter value in (3) to get a proper threshold. Thus, a binary image is obtained as follows:

$$u_{i,j}^b:=\begin{cases}1, & \text{if } \left|u_{i,j}^{m+T}-u_{i,j}^m\right|\ge w\frac{\xi(u^{m+T})+\xi(u^m)}{2}\\ 0, & \text{otherwise}\end{cases},\ \ \forall i\in\{1,...,I\},\ \forall j\in\{1,...,J\},\ w\in(0,3). \tag{7}$$

Then, some morphological operations are applied to the binary image u^b. A dilation process is applied first: $u^b\oplus S$, where S is a $[1\times1]$ square structuring element. Then, a morphological thinning using Zhang-Suen algorithm [20] is performed on the dilated image, its skeleton $Sk\left(u^b\oplus S\right)$ being determined. Other

operations, such as filling the gaps between small connected components that are parts of the same level-line and removing the isolated low-area white spots, are then applied on this thinned image. The processed skeleton, $Sk^p\left(u^b \oplus S\right)$, represents the edge detection result. This method provides a better edge detection than Sobel, Roberts or Prewitt operators and leads to better decompression results even than Canny detector.

Next, the pixels located in the vicinity of the extracted edge points are identified, then coded and stored using a lossless algorithm. Our compression algorithm does not encode the extracted edges, but only the pixels located in their vicinity in the initial $[I \times J]$ image, *Im*. The values from both sides of the detected boundaries are identified determining for each edge point, characterized by $Sk^p\left(u^b \oplus S\right)[i,j] = 1$, the pixels in its neighborhood that do not lie on the edge and have not been already detected. The pixel neighborhood can be based on a 4– or an 8-connectivity. While the 4–neighborhood provides a higher compression ratio, the 8–neighborhood leads to a better decompression result. A lossless RLE-inspired coding algorithm is next applied on the sparse $[I \times J]$ image of these detected *key* pixels, which is equal to *Im* in their locations and 0 elsewhere. This image is transformed into a row vector **R**, the current position in **R** is set at $i = 1$ and initial code vector $\mathbf{C} := \phi$. At each step, one determines the 3-uple $[\mathbf{R}(i), n_i, z_i]$, n_i being the number of successive occurrences of the current value $(\mathbf{R}(i) = ... = \mathbf{R}(i + n_i - 1))$ and z_i is the number of consecutive zeroes after last occurrence $(\mathbf{R}(i + n_i) = \ldots = \mathbf{R}(i + n_i - 1 + z_i) = 0)$. The 3-uple is appended to the code, $\mathbf{C} := [\mathbf{C}, \mathbf{R}(i), n_i, z_i]$, and the current position becomes $i + n_i + z_i$. The final code **C** represents the image compression result.

3. IMAGE DECOMPRESSION USING NONLINEAR DIFFUSION-BASED INPAINTING

The image decompression component of the proposed framework consists of two main processing steps: decoding the compressed edge-related pixels and recovering the entire image from them. A decoding algorithm for the values stored in the code vector is easy to implement. The compressed vector **C** is decoded by expanding $[\mathbf{C}(i), \mathbf{C}(i+1), \mathbf{C}(i+2)]$ into $\left[\underbrace{\mathbf{C}(i), \ldots, \mathbf{C}(i)}_{\mathbf{C}(i+1)}, \underbrace{\mathbf{0} \ldots \mathbf{0}}_{\mathbf{C}(i+2)}\right]$, for each $i = 1, 4, 7, \ldots$, and appending it to **R** (initially void). Final vector **R** of length $I \cdot J$ is then transformed back into a sparse $[I \times J]$ image whose only non-zero pixels are those neighboring the edges of *Im*, let it be denoted I_R. The recovery of the original image from an observed image known only in a few properly selected key pixels, representing the data interpolation points, is performed applying a nonlinear anisotropic diffusion-based structural inpainting technique [5,15]. We propose the next hybrid nonlinear fourth-order PDE interpolation model combining second and fourth order anisotropic diffusions, and having a set of boundary conditions:

$$\begin{cases} \dfrac{\partial u}{\partial t} + \lambda\delta\left(\|\nabla u\|\right)\nabla^2\left(\phi\left(\|\Delta u\|\right)\nabla^2 u\right) - \eta\nabla\cdot\left(\psi\left(\|\nabla u\|\right)\nabla u\right) + \left(1 - 1_\Gamma\right)\left(u - u_0\right) = 0 \\ u(x,y,0) = u_0(x,y), \quad \forall (x,y) \in \Omega \\ \dfrac{\partial u}{\partial \vec{n}} = 0 \\ u(t,x,y) = 0, \quad \forall (x,y) \in \partial\Omega \end{cases} \tag{8}$$

where $\lambda \in [1.4, 2)$, $\eta \in (0.5, 1]$, $\Gamma \subset \Omega \subseteq R^2$ represents the inpainting region corresponding to the missing parts, the observed image is a partial two-dimensional function $u_0 : \Omega \setminus \Gamma \to R$, the diffusivity function ψ of the second-order anisotropic diffusion term is that used by the PDE-based edge segmentation model and defined in (2), the function δ related to the term controlling the diffusion speed is that provided by (4) and the positive function corresponding to the fourth-order nonlinear diffusion-based component is modeled as:

$$\varphi:[0,\infty)\to[0,\infty),\quad \varphi(s)=\gamma\sqrt[k]{\frac{\xi(u)}{\varsigma\left|\xi(u)+s\right|^{k+1}+\tau}} \tag{9}$$

where $\tau\in(2,4]$, $\gamma,\varsigma\in[0.5,1)$, $k\in\{2,\ldots,5\}$ and ξ is the conductance parameter given by (3). The inpainting mask of this reconstruction model is provided by characteristic function $1_\Gamma(x,y)=\begin{cases}1, & \text{if } (x,y)\in\Gamma\\ 0, & \text{if } (x,y)\notin\Gamma\end{cases}$. The fourth-order nonlinear diffusion-based inpainting scheme (8) performs a successful completion of the evolving image given by the function $u:\Omega\to R:u|_{\Omega\setminus\Gamma}=u_0$ by directing the diffusion process mostly to the inpainting region, using the inpainting mask and employing the information of the surrounding known zones.

The behavior of this hybrid PDE-based interpolation system is controlled by varying the values of the λ and η parameters. It comes close to a second-order PDE inpainting model if η is set high and λ gets a low value, which provides a faster reconstruction but a recovered image not so close to the original, for our set of known pixels. For a high λ and a low η value, it gets close to a fourth-order PDE inpainting scheme that provides more natural recovered images but requires a longer execution time. Therefore, a successful interpolation process should use the both nonlinear diffusion-based components, with some properly selected parameters. We have selected above some proper intervals for these coefficients, which lead to good reconstruction results.

The proposed fourth-order PDE scheme is well-suited for structural interpolation but not for texture-based inpainting. Since the model does not inpaint properly the textures, our compression technique works successfully for non-textured images only. Also, this PDE model is well-posed, admitting a unique weak solution representing the recovered image and computed applying a numerical approximation algorithm that is constructed for the model (8) using the finite difference method [19]. The quantization of space and time coordinates provided in (5) is used here, too. The differential equation in (8) can be written as:

$$\frac{\partial u}{\partial t}+(1-1_\Gamma)(u-u_0)=\eta\left(\frac{\partial}{\partial x}\left(\psi\left(|\nabla u|\right)u_x\right)+\frac{\partial}{\partial y}\left(\psi\left(|\nabla u|\right)u_y\right)\right)-\lambda\delta\left(\|\nabla u\|\right)\nabla^2\left(\varphi\left(\|\nabla u\|\right)\nabla^2 u\right). \tag{10}$$

Its left term is approximated as $\frac{u_{i,j}^{n+\Delta t}-u_{i,j}^n}{\Delta t}+(1-1_\Gamma)\left(u_{i,j}^n-u_{i,j}^0\right)=u_{i,j}^{n+\Delta t}\frac{1}{\Delta t}+u_{i,j}^n\left(1-1_\Gamma-\frac{1}{\Delta t}\right)-u_{i,j}^0\left(1-1_\Gamma\right)$, which becomes $u_{i,j}^{n+1}-1_\Gamma u_{i,j}^n+u_{i,j}^0(1_\Gamma-1)$ for $h=\Delta t=1$. The second-order diffusion-based term, $\eta\left(\frac{\partial}{\partial x}\left(\psi\left(|\nabla u|\right)u_x\right)+\frac{\partial}{\partial y}\left(\psi\left(|\nabla u|\right)u_y\right)\right)$, is discretized as described in §2.2 and the approximation $\eta\left(\psi_{i+\frac{1}{2},j}\left(u_{i+1,j}^n-u_{i,j}^n\right)-\psi_{i-\frac{h}{2},j}\left(u_{i,j}^n-u_{i-1,j}^n\right)+\psi_{i,j+\frac{1}{2}}\left(u_{i,j+1}^n-u_{i,j}^n\right)-\psi_{i,j-\frac{1}{2}}\left(u_{i,j}^n-u_{i,j-1}^n\right)\right)$ is obtained. The fourth-order diffusion component, $\lambda\delta\left(\|\nabla u\|\right)\nabla^2\left(\varphi\left(\|\nabla u\|\right)\nabla^2 u\right)$, is approximated next, using the finite difference-based Laplacian discretization [19] and the discretization $\delta_{i,j}=\delta\left(\left\|u_{i,j}\right\|\right)$ explained also in §2.2. Thus, we get:

$$\lambda\delta_{i,j}\Delta\varphi_{i,j}=\lambda\delta_{i,j}\frac{\varphi_{i+1,j}+\varphi_{i-1,j}+\varphi_{i,j+1}+\varphi_{i,j-1}-4\varphi_{i,j}}{h^2}, \tag{11}$$

where $\varphi_{i,j}=\varphi\left(\left|\Delta u_{i,j}\right|\right)\nabla^2 u_{i,j}$ and $\Delta u_{i,j}=\nabla^2 u_{i,j}=\frac{u^n_{i+h,j}+u^n_{i-h,j}+u^n_{i,j+h}+u^n_{i,j-h}-4u_{i,j}}{h^2}$.

Using the above discretizations for $h=\Delta t=1$ one obtains the explicit numerical approximation scheme:

$$u_{i,j}^{n+1} = u_{i,j}^{n}\left(1_\Gamma - \eta\left(\psi_{i+\frac{1}{2},j} + \psi_{i-\frac{1}{2},j} + \psi_{i,j+\frac{1}{2}} + \psi_{i,j-\frac{1}{2}}\right)\right) + \eta\left(u_{i+1,j}^{n}\psi_{i+\frac{1}{2},j} + u_{i-1,j}^{n}\psi_{i-\frac{1}{2},j} + u_{i,j+1}^{n}\psi_{i,j+\frac{1}{2}} + u_{i,j-1}^{n}\psi_{i,j-\frac{1}{2}}\right) +$$
$$+ u_{i,j}^{0}\left(1 - 1_\Gamma\right) - \lambda\delta_{i,j}\left(\varphi_{i+1,j} + \varphi_{i-1,j} + \varphi_{i,j+1} + \varphi_{i,j-1} - 4\varphi_{i,j}\right). \quad (12)$$

This iterative numerical approximation algorithm is consistent to the nonlinear fourth-order PDE model (8) and converges to its solution. It starts with the image of the pixels adjacent to edges as the initial image, therefore $u_{i,j}^{0} = I_R(i,j),\ \forall i \in \{1,\ldots,I\},\ j \in \{1,\ldots,J\}$. The iterative inpainting algorithm (12) is applied on the current image, u^n, for each $n = 0,1,\ldots,N$. The final interpolation output, u^{N+1}, represents the reconstructed image and the lossy decompression result.

The proposed iterative scheme provides effective inpainting results but does not execute fast for this image data. Thus, it requires a high number of iterations, which means a large *N* value depending also on the image size, to achieve an image that is similar to the original *Im*, from I_R, whose most pixel values are 0. The decompressed image u^{N+1} is not identical to *Im*, given the loss of information provided by our diffusion-based compression method, but it is visually similar.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

The described PDE-based (de)/compression approach has been tested successfully on hundreds of digital images, in both clean and noisy conditions. It achieves good values for the performance measures of the data compression, such as the compression rate (in bits per pixel), compression ratio, fidelity and quality.

Our technique may obtain higher compression ratios by varying some parameters of its components, but the fidelity and quality measures could be negatively altered. For example, increasing the threshold in (7), by using a higher value for *w*, would lead to fewer detected edge points, which means fewer key pixels, therefore a higher compression ratio. Using the 4-connectivity instead of the 8-connectivity in the key pixel detection process would lead to a higher ratio, too.

Another way to improve the compression rates and ratios is to transform the lossless pixel encoding algorithm described in §2.2 into a lossy coding procedure, by replacing the equality $\mathbf{R}(i) = \ldots = \mathbf{R}(i+n_i-1)$ to a inequality $\left\|\mathbf{R}(i+j) - \mathbf{R}(i+j+1)\right\| \le \text{threshold},\ \forall j \in [0, n_i - 2]$. As the compression ratio is increasing, the quality and fidelity of the decompressed image are decreasing. These performance measures are evaluated using similarity metrics such as the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [21]. The decompressed versions of the images coded by our technique get good values of these metrics when measured against originals. The compression rates obtained from our tests lie between 0.40 and 0.75 bpp, corresponding to compression ratios from 19:1 down to 11:1.

The stages of an image compression process are described in Fig. 1. Original $[435 \times 328]$ *Peppers* image displayed in (a) is segmented and its edge-image, $Sk^p\left(u^b \oplus S\right)$, is represented in (b). The key pixels of I_R, obtained with 4-connectivity and displayed in (c), are encoded into a code $\mathbf{C}$ of length 12195. A 12:1 ratio is achieved for this image by our technique. While the coding process executes fast, the decoding part runs much slower and requires more time, given the complexity of the PDE inpainting-based decompression scheme. The decompressed image obtained from $\mathbf{C}$ and depicted in (d) has a good quality and fidelity, getting a PSNR = 29.7 (dB) for this compression ratio, but it is reached after many iterations of the scheme (12), $N = 1950$.

Method comparison has been also performed. Our data interpolation point selection, considering only the pixels along the essential level lines, has proved to be a better solution than other scattered data point selection methods, like the randomly point selection or BTTC. Also, while not being faster, the proposed anisotropic diffusion interpolation technique provides better results than other PDE inpainting solutions used for coding.

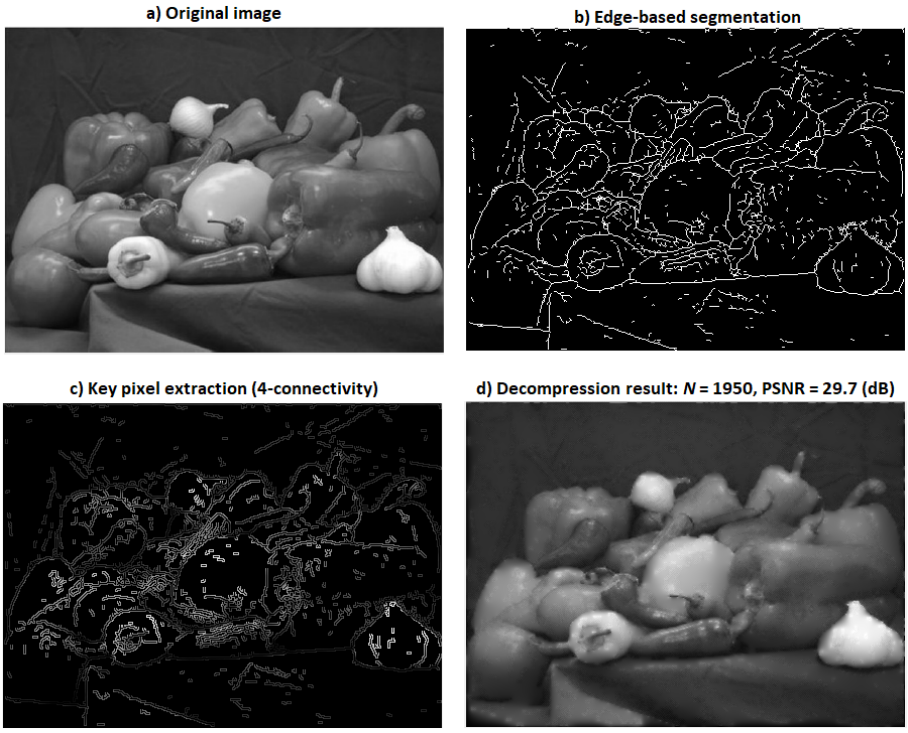

Fig. 1 – Stages of an image compression process, with the compression ratio = 12:1, using the proposed method.

Fig. 2 – Compression results of our AD scheme and other PDE and non-PDE methods at compression rate = 0.4 bpp.

We have assessed the performances of various compression techniques, by applying PSNR and SSIM for them at the same compression rates, and found that our anisotropic diffusion framework outperforms the PDE-based algorithms using random image sparsification and some compression schemes using BTTC with PDE models like linear homogeneous diffusion, biharmonic smoothing or EED. It also outperforms BTTC-L model [10] and provides better compression results than JPEG at higher rates, but it is outperformed by JPEG 2000 and R-EED codec.

Some method comparison results are described in Fig. 2 and Table 1. Figure 2 describes the output of several compression techniques on the $[256 \times 256]$ *Trui* image, at a rate of 0.40 bpp. The proposed AD-based method provides a good compression result, displayed in (g). Its output is better than those of the most PDE and non-PDE compression methods considered here, as it results also from Table 1, where only R-EED and JPEG 2000 provide higher PSNR and SSIM values than it.

Table 1

PSNR and SSIM values achieved by several techniques at a compression rate of 0.4 bpp

Compression Technique	**PSNR**	**SSIM**
The proposed AD-based technique	27.8777 (dB)	0.8231
Linear homogeneous diffusion	20.8971 (dB)	0.6866
Biharmonic smoothing operator	21.4235 (dB)	0.7230
BTTC-L compression scheme	22.8565 (dB)	0.7487
BTTC-EED compression	25.6607 (dB)	0.7738
R-EED codec	28.9994 (dB)	0.8586
JPEG	27.7475 (dB)	0.8143
JPEG 2000	28.4545 (dB)	0.8467

5. CONCLUSIONS

A nonlinear anisotropic diffusion-based lossy image compression and decompression technique using edge-based segmentation has been described in this paper. The proposed framework uses novel PDE-based image analysis solutions, representing combined anisotropic diffusion models, for both compression and decompression. Some consistent finite difference based numerical approximations are constructed for both models.

Given its strong edge defining character, the first anisotropic diffusion scheme, combining a second-order nonlinear PDE to a classic additive noise filtering kernel, is successfully used within the proposed edge extraction method that is also based on some mathematical morphological operations and outperforms many state of the art edge detectors. Although the achieved edge information is essential for the compression component, the proposed coding algorithm does not encode the edges, but only the *key* pixels neighboring them and representing the scattered data interpolation points, using a new lossless method. Our compression module, composed of the edge detector and the key pixel detection and coding algorithms, provides good compression rates and ratios, in both clean and noisy conditions.

The decompression module of the framework, consisting of a key pixel decoder and a PDE-based pixel interpolation scheme, achieves good quality and fidelity for decompressed images but does not operate fast and its execution time may reach several minutes for large images, due to the complexity of the second anisotropic diffusion-based model of the framework, representing another contribution of our work. The nonlinear fourth-order PDE structural interpolation model, combining second and fourth order diffusions, inpaints properly the sparse images of key pixels, but applying many interations of its finite difference-based numerical approximation scheme. We have been interested more in a good decompression output than in a low running time when developing this technique. Also, given its structure-based inpainting scheme, it can decompress properly only non-textured images.

Some successful tests and method comparisons illustrating its effectiveness have been also described. The proposed edge-based anisotropic diffusion approach provides better decompressed images than many conventional and PDE-based coding methods, at the same compression rate. Our future research will focus

on improving this technique, by finding a solution to fasten the convergence of the approximation algorithm of the nonlinear diffusion-based interpolation model and to adapt it for the texture inpainting also.

REFERENCES

1. R. GONZALES, R. WOODS, *Digital image processing*, Prentice Hall, New York, NY, USA, 2nd edition, 2001.
2. K. SAYOOD, *Introduction to data compression*, Morgan Kaufmann, Third edition, 2005.
3. V. BHASKARAN, K. KONSTANTINIDES, *Image and video compression standards*, Kluwer Academic Press, Boston, 1995.
4. G. AUBERT, P. KORNPROBS, *Mathematical problems in image processing: partial differential equations and the calculus of variations*, Vol. 147, Springer Science & Business Media, 2006.
5. C. B. SCHONLIEB, *Partial differential equation methods for image inpainting*, Vol. 29, Cambridge University Press, 2015.
6. D. MUMFORD, J. SHAH, *Optimal approximation by piecewise smooth functions and associated variational problems*, Comm. Pure Appl. Math., **42**, pp. 577–685, 1989.
7. J. ZHANG, K. CHEN, *Variational image registration by a total fractional-order variation model*, Journal of Computational Physics, **293**, pp. 442–461, 2015.
8. P. T. PETER, *Understanding and advancing PDE-based image compression*, PhD Thesis, Saarland University, 2016.
9. I. GALIC, J. WEICKERT, M. WELK, A. BRUHN, A. BELYAEV, H.-P. SEIDEL, *Image compression with anisotropic diffusion*, Journal of Mathematical Imaging and Vision, **31**, *2-3*, pp. 255–269, 2008.
10. R. DISTASI, M. NAPPI, S. VITULANO, *Image compression by B-tree triangular coding*, IEEE Transactions on Communications, **45**, *9*, pp. 1095–1100, 1997.
11. C. SCHMALTZ, J. WEICKERT, A. BRUHN, *Beating the quality of JPEG 2000 with anisotropic diffusion*, in: *Pattern Recognition*, Lecture Notes in Computer Science, Vol. 5748, pp. 452–461, Springer, Berlin, 2009.
12. M. MAINBERGER, J. WEICKERT, *Edge-based image compression with homogeneous diffusion*, in: *Computer Analysis of Images and Patterns,* Lecture Notes in Computer Science, Vol. 5702, pp. 476–483, Springer, Berlin, 2009.
13. J. CHEN, F. YE, J. DI, C. LIU, A. MEN, *Depth map compression via edge-based inpainting*, Proceedings of the 29th Picture Coding Symposium, pp. 57–60, Kraków, Poland, May 2012.
14. T. BARBU, A. FAVINI, *Rigorous mathematical investigation of a nonlinear anisotropic diffusion-based image restoration model*, Electronic Journal of Differential Equations, *129*, pp. 1–9, 2014.
15. T. BARBU, *Variational image inpainting technique based on nonlinear second-order diffusions*, Computers & Electrical Engineering, **54**, pp. 345–353, 2016.
16. T. BARBU, *Robust contour tracking model using a variational level-set algorithm*, Numerical Functional Analysis and Optimization, **35**, *3*, pp. 263–274, 2014.
17. J. CANNY, *A computational approach to edge detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, **8**, pp. 679–714, 1986.
18. D. MARR, E. HILDRETH, *Theory of edge detection*, Proceedings of the Royal Society of London. Series B, Biological Sciences, **207**, *1167*, pp. 187–217, 1980.
19. P. JOHNSON, *Finite Difference for PDEs*, School of Mathematics, University of Manchester, Semester I, 2008.
20. T. Y. ZHANG, C. Y. SUEN, *A fast parallel algorithm for thinning digital patterns*, Commun. ACM, **27**, pp. 236–239, 1984.
21. K.-H. THUNG, P. RAVEENDRAN, *A survey of image quality measures*, International Conference for Technical Postgraduates (TECHPOS), pp. 1–4, Kuala Lumpur, Malaysia, December 2009.

Received November 7, 2018