A Geometric PDE for Interpolation of M-channel Data

Frank Lenzen[1] and Otmar Scherzer[1 2]

[1] Department of Mathematics, University of Innsbruck,
Technikerstrasse 21a, A-6020 Innsbruck, Austria,
Frank.Lenzen,Otmar.Scherzer@uibk.ac.at,
WWW home page: http://infmath.uibk.ac.at
[2] Johann Radon Institute for Computational and
Applied Mathematics (RICAM),
Austrian Academy of Sciences,
Altenbergerstrasse 69, A-4040 Linz, Austria

Abstract. We propose a partial differential equation to be used for interpolating M-channel data, such as digital color images. This equation is derived via a semi-group from a variational regularization method for minimizing displacement errors. For actual image interpolation, the solution of the PDE is projected onto a space of functions satisfying interpolation constraints. A comparison of the test results with standard and state-of-the-art interpolation algorithms shows the competitiveness of this approach.

1 Introduction

A frequent task in image processing is *interpolation*, which we refer to as the process of assigning a discrete set of pixel positions and according discrete M-channel image data (e.g. RGB color data) an interpolating *function*. Interpolation is frequently used for *zooming into* or *scaling* digital images. A special kind of image interpolation problems is *inpainting*, i.e. the problem of reconstructing lost or corrupted parts of images.

Linear interpolation (that is convolution methods) [18], such as for example nearest neighbor, spline, and the Whittaker-Shannon interpolation [14, 4], is computationally efficient but produce unpleasant artifacts. On the other hand, *nonlinear* methods adapting to geometrical structures can produce more visually attractive results but are computationally more demanding . Nowadays, most of these nonlinear methods are motivated by energy minimization or by scale spaces of partial differential equations, see for example [1, 22, 21, 18]. In particular for inpainting such nonlinear methods are widely used, see for example [2, 5, 6, 23].

In this paper we derive a partial differential equation that is designed to correct and filter for displacement errors in M-channel data. Combined with the interpolation ideas of [11, 16], this method is suited for interpolation.

The paper is organized as follows: In Section 2 we consider a variational ansatz for correcting displacement errors. Application of the semi-group concepts yields a PDE, which can be considered the gradient flow of the variational problem. A relationship of our PDE to the *Mean Curvature Flow* (MCF) equation is established. Our approach is combined with interpolation constraints in Section 3.

For comparison, we show in Section 4 results from the proposed method and from interpolation methods from the scale space literature. In particular we take into account the GREYCstoration software of Tschumperlé [21] and the interpolation method proposed by Roussos and Maragos [18, 19]. The paper ends with a conclusion in Section 5.

2 Displacement Regularization

Let $\mathbf{u} : \Omega \to \mathbb{R}^M$ be an M channel function representing *continuous* M-channel data on a bounded open domain $\Omega \subseteq \mathbb{R}^2$.

We presume the following image acquisition model: Data $\mathbf{u}^{(0)}$ of $\mathbf{u}$ are given, which satisfy

$$\mathbf{u}^{(0)} = \mathbf{u} \circ \mathbf{\Phi} \,, \tag{1}$$

where $\mathbf{\Phi} : \Omega \to \Omega$ is a displacement vector field.

In the following we consider the problem of finding $(\mathbf{u}, \mathbf{\Phi})$ satisfying (1) such that the displacement $\mathbf{\Phi} - \mathrm{Id}$ is small and u has minimal total variation. A variational method corresponding to this problem consists in minimization of

$$\frac{1}{2} \int_\Omega |\mathbf{\Phi}(\mathbf{x}) - \mathbf{x}|^2 \, d\mathbf{x} + \alpha \int_\Omega |\nabla \mathbf{u}(\mathbf{x})| \, d\mathbf{x} \,, \tag{2}$$

for small $\alpha > 0$ over the set of functions satisfying $\mathbf{u}^{(0)} = \mathbf{u} \circ \mathbf{\Phi}$. Here

$$\nabla u = \begin{pmatrix} \partial_1 u_1 & \partial_1 u_2 & \partial_1 u_3 \\ \partial_2 u_1 & \partial_2 u_2 & \partial_2 u_3 \end{pmatrix} \text{ and } |\nabla \mathbf{u}(\mathbf{x})| = \left(\sum_{j=1}^{M} \sum_{i=1}^{2} (\partial_i u_j(\mathbf{x}))^2 \right)^{1/2} .$$

We want to avoid solving a coupled system for $(\mathbf{u}, \mathbf{\Phi})$, and therefore we assume that $\mathbf{u}$ is a smooth function, so that we can make a first order Taylor series expansion. Then it follows from our modeling assumptions that

$$\mathbf{u}^{(0)}(\mathbf{x}) = (\mathbf{u} \circ \mathbf{\Phi})(\mathbf{x}) = \mathbf{u}(\mathbf{x} + (\mathbf{\Phi}(\mathbf{x}) - \mathbf{x})) \approx \mathbf{u}(\mathbf{x}) + \nabla \mathbf{u}^T(\mathbf{x}) \, (\mathbf{\Phi}(\mathbf{x}) - \mathbf{x}) \,. \tag{3}$$

Here, $\approx$ symbolizes that the left hand side approximates the right hand side for small displacements $\mathbf{\Phi} - \mathrm{Id}$. In the following, we assume that equality holds instead of $\approx$, which implies that only small displacements occur.

Note that the equation $\nabla \mathbf{u}^T(\mathbf{x})(\mathbf{\Phi}(\mathbf{x}) - \mathbf{x}) = \mathbf{u}^{(0)}(\mathbf{x}) - \mathbf{u}(\mathbf{x})$ for unknown $\mathbf{\Phi}(\mathbf{x}) - \mathbf{x}$ is overdetermined. In case that the difference $\mathbf{u}^{(0)}(\mathbf{x}) - \mathbf{u}(\mathbf{x})$ is not only caused by a distortion $\mathbf{\Phi}$, no solution to this problem might exist. To overcome this problem, we consider the minimization of

$$\left|\nabla \mathbf{u}^T(\mathbf{x})(\mathbf{\Phi}(\mathbf{x}) - \mathbf{x}) - \mathbf{u}^{(0)}(\mathbf{x}) + \mathbf{u}(\mathbf{x})\right|^2 , \tag{4}$$

that is, we search for the displacement vector $\mathbf{\Phi}(\mathbf{x}) - \mathbf{x}$, which fits best to the data $(\mathbf{u}^{(0)}(\mathbf{x}), \mathbf{u}(\mathbf{x}))$. The minimizer of (4) is given by

$$\mathbf{\Phi}(\mathbf{x}) - \mathbf{x} = (\nabla \mathbf{u}^T(\mathbf{x}))^\dagger(\mathbf{u}^{(0)}(\mathbf{x}) - \mathbf{u}(\mathbf{x})), \tag{5}$$

where $(\nabla \mathbf{u}(\mathbf{x}))^\dagger$ denotes the *pseudo–inverse* (see [17]) of $\nabla \mathbf{u}(\mathbf{x})$.

For notational convenience, we leave out the dependence of $\mathbf{u}$ with respect to $\mathbf{x}$ in the following. Inserting (5) into (2) gives the functional

$$\mathcal{F}^0_{\mathbf{u}^{(0)}}(\mathbf{u}) := \frac{1}{2}\int_\Omega (\mathbf{u} - \mathbf{u}^{(0)})^T(\nabla \mathbf{u}^T \nabla \mathbf{u})^\dagger(\mathbf{u} - \mathbf{u}^{(0)}) + \alpha \left|\nabla \mathbf{u}\right| \, d\mathbf{x} \, . \tag{6}$$

In order to avoid computation of the pseudo–inverse, we additionally regularize the probably singular matrix $\nabla \mathbf{u}^T \nabla \mathbf{u}$ by the regular, symmetric, and strictly positive definite matrix $(\varepsilon I + \nabla \mathbf{u}^T \nabla \mathbf{u})$ with some $\varepsilon > 0$. To summarize, we consider in the sequel the variational problem of minimizing

$$\mathcal{F}^\varepsilon_{\mathbf{u}^{(0)}}(\mathbf{u}) := \frac{1}{2}\int_\Omega (\mathbf{u} - \mathbf{u}^{(0)})^T(\varepsilon I + \nabla \mathbf{u}^T \nabla \mathbf{u})^{-1}(\mathbf{u} - \mathbf{u}^{(0)}) + \alpha \left|\nabla \mathbf{u}\right| \, d\mathbf{x} \, . \tag{7}$$

For this functional, existence theory within the classical framework of the *Calculus of Variations* [7, 8] is not applicable. Moreover for a theoretical analysis, minimization has in fact to be considered over the space of M-channel functions with components of finite total variation. In order to implement the minimization of $\mathcal{F}^\varepsilon_v$ numerically, quasi-convexification techniques would be most efficient. This approach requires the analytical calculation of the quasi-convex envelope of the function

$$(\mathbf{x}, \xi, \nu) \to \frac{1}{2}(\xi - v(\mathbf{x}))^T(\varepsilon I + \nu^T \nu)^{-1}(\xi - v(\mathbf{x})) + \alpha \left|\nu\right|$$

with respect to ν. However, the quasi-convex envelope function is not known so far, and thus efficient numerical minimization based on this approach is not at hand.

In the following we recall the convex semi-group solution concept [3]: Let $\mathcal{R} : H \to \mathbb{R} \cup \{\infty\}$ be a convex functional on a Hilbert space H, and let $\mathbf{u}_\alpha$ be a minimizer of the variational regularization functional

$$\mathcal{G}_{\mathbf{u}^{(0)}}(\mathbf{u}) := \frac{1}{2}\left\|\mathbf{u} - \mathbf{u}^{(0)}\right\|^2_H + \alpha \mathcal{R}(\mathbf{u}) \, .$$

Then, for $\mathbf{u}^{(0)}$ sufficiently smooth, $(\mathbf{u}_\alpha - \mathbf{u}^{(0)})/\alpha$ converges for $\alpha \to 0$ to an element in the subgradient $\partial\mathcal{R}(\mathbf{u}^{(0)})$ of $\mathcal{R}$. Choosing $\mathbf{u}^{(k)} \in \operatorname{argmin} \mathcal{G}_{\mathbf{u}^{(k-1)}}$, iterative minimization of $\mathcal{G}_{\mathbf{u}^{(k)}}$ yields an approximation of the solution of the flow

$$\frac{\partial \mathbf{u}}{\partial t} \in \partial \mathcal{R}(\mathbf{u})$$

at scale $t = k\alpha$. In other words, variational regularization approximates a diffusion filtering scale space, which is the associated gradient flow equation. For convex semi-groups the solutions of diffusion filtering and variational methods are comparable and look rather similar [20].

We expect a similar behavior for the non-convex functional $\mathcal{F}^{\varepsilon}_{\mathbf{u}^{(0)}}$ and derive the according flow equation, which is the gradient flow associated with (7). We use the abbreviations

$$\begin{aligned} A^{\varepsilon}(\mathbf{u}) &:= \left(\varepsilon I + \nabla \mathbf{u}^T \nabla \mathbf{u}\right)^{-1} \quad \text{and} \\ S^{\varepsilon}_{\mathbf{u}^{(k-1)}}(\mathbf{u}) &:= \frac{1}{2} \int_{\Omega} (\mathbf{u} - \mathbf{u}^{(k-1)})^T A^{\varepsilon}(\mathbf{u})(\mathbf{u} - \mathbf{u}^{(k-1)}) \, d\mathbf{x} \, . \end{aligned}$$

The *directional derivative* of $S^{\varepsilon}_{\mathbf{u}^{(k-1)}}$ at $\mathbf{u}$ in direction ϕ (provided it exists) satisfies

$$\begin{aligned} \partial_{\tau} S^{\varepsilon}_{\mathbf{u}^{(k-1)}}(\mathbf{u} + \tau\phi) = &\int_{\Omega} \phi^T A^{\varepsilon}(\mathbf{u})(\mathbf{u} - \mathbf{u}^{(k-1)}) \, d\mathbf{x} + \\ &\frac{1}{2} \int_{\Omega} (\mathbf{u} - \mathbf{u}^{(k-1)})^T \partial_{\mathbf{u},\phi} A^{\varepsilon}(\mathbf{u}) \, (\mathbf{u} - \mathbf{u}^{(k-1)}) \, d\mathbf{x} \, , \end{aligned} \tag{8}$$

where

$$\partial_{\mathbf{u},\phi} A^{\varepsilon}(\mathbf{u}) := \lim_{\tau \to 0} \frac{A^{\varepsilon}(\mathbf{u} + \tau\phi) - A^{\varepsilon}(\mathbf{u})}{\tau} .$$

In a similar way, the directional derivative of $R_{\alpha}(\mathbf{u}) := \alpha \int_{\Omega} |\nabla \mathbf{u}|$ at $\mathbf{u}$ in direction ϕ can be derived in a formal way:

$$\partial_{\tau} R_{\alpha}(\mathbf{u} + \tau\phi) = \alpha \int_{\Omega} \nabla \phi^T \frac{\nabla \mathbf{u}}{|\nabla \mathbf{u}|} \, d\mathbf{x}. \tag{9}$$

Note that the right hand side of (9) is meant as the subdifferential of the TV semi-norm evaluated in the direction of ϕ.

Using (8) and (9), the optimality condition for the minimizer $\mathbf{u}^{(k)}$ of $\mathcal{F}^{\varepsilon}_{\mathbf{u}^{(k-1)}}$ reads as

$$\begin{aligned} &\int_{\Omega} \phi^T A^{\varepsilon}(\mathbf{u}^{(k)}) \frac{\mathbf{u}^{(k)} - \mathbf{u}^{(k-1)}}{\alpha} \, d\mathbf{x} \\ +&\frac{1}{2} \int_{\Omega} \frac{(\mathbf{u}^{(k)} - \mathbf{u}^{(k-1)})^T}{\alpha} \partial_{\mathbf{u}^{(k)},\phi} A^{\varepsilon}(\mathbf{u}^{(k)}) \, (\mathbf{u}^{(k)} - \mathbf{u}^{(k-1)}) \, d\mathbf{x} \\ =& -\int_{\Omega} \nabla \phi^T \frac{\nabla \mathbf{u}^{(k)}}{|\nabla \mathbf{u}^{(k)}|} \, d\mathbf{x}. \end{aligned} \tag{10}$$

Let $t > 0$ be fixed and $k = \lfloor t/\alpha \rfloor$, then, as in the convex case, we can expect that $(\mathbf{u}^{(k)} - \mathbf{u}^{(k-1)})/\alpha$ converges to $\partial_t \mathbf{u}(t)$ for $\alpha \to 0$. From this it follows then that $\mathbf{u}^{(k)} - \mathbf{u}^{(k-1)} \to 0$, and from (10) it follows that

$$\int_{\Omega} \phi^T A^{\varepsilon}(\mathbf{u}(t)) \partial_t \mathbf{u}(t) \, d\mathbf{x} = -\int_{\Omega} \nabla \phi^T \frac{\nabla \mathbf{u}(t)}{|\nabla \mathbf{u}(t)|} \, d\mathbf{x}. \tag{11}$$

Using Green's formula and the fundamental lemma, from (11) the strong formulation

$$A^{\varepsilon}(\mathbf{u}(t))\partial_t \mathbf{u}(t) = \left(\varepsilon I + \nabla \mathbf{u}^T(t) \nabla \mathbf{u}(t)\right)^{-1} \partial_t \mathbf{u}(t) = \nabla \cdot \left(\frac{\nabla \mathbf{u}(t)}{|\nabla \mathbf{u}(t)|} \right) , \tag{12}$$

follows, where $\mathbf{u}(t)$ satisfies natural (Neumann) boundary conditions.

In the following, we leave out the dependence of $\mathbf{u}$ with respect to t for notational convenience. Multiplying both sides of (12) by $(\varepsilon I + \nabla \mathbf{u}^T \nabla \mathbf{u})$, we get

$$\partial_t \mathbf{u} = (\varepsilon I + \nabla \mathbf{u}^T \nabla \mathbf{u}) \; \nabla \cdot \left(\frac{\nabla \mathbf{u}}{|\nabla \mathbf{u}|} \right) . \tag{13}$$

Moreover, the initial condition associated with the flow is $u(0) := u^{(0)}$. Now, letting $\varepsilon \to 0$, which only seems to make sense mathematically if $M \leq 2$, we obtain the evolutionary partial differential equation

$$\partial_t \mathbf{u} = (\nabla \mathbf{u}^T \nabla \mathbf{u}) \; \nabla \cdot \left(\frac{\nabla \mathbf{u}}{|\nabla \mathbf{u}|} \right) . \tag{14}$$

Remark 1. For scalar data ($M = 1$) the equation (14) reads as

$$\partial_t u = |\nabla u|^2 \; \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) . \tag{15}$$

One recognizes that (15) differs from the Mean Curvature Flow equation by the leading factor $|\nabla u|^2$ instead of $|\nabla u|$.

We generalize the functional in (6) to

$$\frac{1}{2} \int_{\Omega} (\mathbf{u} - \mathbf{u}^{(0)})^T \left((\nabla \mathbf{u}^T \nabla \mathbf{u})^p \right)^{\dagger} (\mathbf{u} - \mathbf{u}^{(0)}) + \alpha \, |\nabla \mathbf{u}| \; d\mathbf{x} \tag{16}$$

with $p \geq 0$. We note that the power of a matrix is defined via spectral decomposition. The case $p = 1/2$ is of particular interest, because

- the functional (16) becomes invariant under affine rescaling of the image brightness.
- The semi-group approach (see also [10] for the scalar case) gives the gradient flow

  $$\partial_t \mathbf{u} = (\nabla \mathbf{u}^T \nabla \mathbf{u})^{\frac{1}{2}} \; \nabla \cdot \left(\frac{\nabla \mathbf{u}}{|\nabla \mathbf{u}|} \right) ,$$

  which, in the scalar case, is the Mean Curvature Flow equation. For an analytical comparison of the solution of (16) for scalar, radial-symmetric monotonous data to the MCF solution we refer to [9].

3 Interpolation of M-Channel Data

The evolution equation (14) can be used for interpolating discrete M-channel data by restricting $\mathbf{u}$ to satisfy interpolation constraints. The problem of interpolating M-channel data has already been studied in the literature before, see for example [1, 21, 18, 19]. The difference between the approaches by [21, 18, 19] and our approach are the different PDEs for filtering: [21, 18, 19] use *anisotropic diffusion*, whereas the PDE (14) generalizes the *Mean Curvature Flow* equation.

To begin with, we recall the interpolation constraints proposed in [11, 16]. For the simplicity of notation we restrict ourself to M-channel data defined on a two-dimensional rectangular domain

$$\Omega := \left(\frac{1}{2}, N_x + \frac{1}{2}\right) \times \left(\frac{1}{2}, N_y + \frac{1}{2}\right),$$

where $N_x, N_y \in \mathbb{N}$. The domain is partitioned into cells ('pixels')

$$Q_{i,j} := \left(i - \frac{1}{2}, i + \frac{1}{2}\right) \times \left(j - \frac{1}{2}, j + \frac{1}{2}\right), \quad (i,j) = (1,1), (1,2) \ldots, (N_x, N_y) .$$

Let G be a kernel function defined on $\mathbb{R}^2$ and compactly supported in $[-\frac{1}{2}, \frac{1}{2}]^2$. Let $\mathbf{Z} := (z_{m,i,j})$ a tensor, which denotes sampled data of a function $G * \mathbf{u} : \mathbb{R}^2 \to \mathbb{R}^M$ at the positions (i,j). Here $*$ denotes the convolution operator. In particular:

$$z_{m,i,j} := (G * u_m)(i,j), \quad (m,i,j) = (1,1,1), (1,1,2) \ldots, (M, N_x, N_y) . \tag{17}$$

Examples for kernel functions typically used in literature are listed in [18].

We rewrite (17) as follows: Let $G_{i,j} := G(\cdot - (i,j))$, then

$$z_{m,i,j} = \langle G_{i,j}, u_m \rangle_{L^2(\Omega)}, \quad (m,i,j) = (1,1,1), \ldots, (M, N_x, N_y) .$$

We say that an M-channel function $\mathbf{u} = (u^1, \ldots, u^M)$ satisfies the interpolation constraints for some discrete data $\mathbf{Z} = (z_{m,i,j})$, if

$$\langle G_{i,j}, u_m \rangle_{L^2(\Omega)} = z_{m,i,j}.$$

The set of functions satisfying the interpolation constraints for data $\mathbf{Z}$ is denoted by $\mathcal{U}_{\mathbf{Z},G}$.

Example 1. We consider for G the two-dimensional δ distribution, i.e.,

$$G(x,y) = \delta(x)\delta(y).$$

Then $z_{m,i,j} = u_m((i,j))$. The *nearest neighbor (componentwise, piecewise constant) interpolation* reads as

$$u_m^{(0)}|_{Q_{i,j}} = z_{m,i,j}, \qquad (m,i,j) = (1,1,1), \ldots, (M, N_x, N_y) .$$

Here, $\mathbf{u}^{(0)} = \mathbf{u} \circ \mathbf{\Phi}$, where $\mathbf{\Phi}(x,y)|_{Q_{i,j}} = (i,j)$. In particular $\mathbf{u}$ can be interpreted as a distortion of $\mathbf{u}^{(0)}$ by a local sampling displacement $\mathbf{\Phi}$.

Now let $\mathbf{u}^{(0)} \in \mathcal{U}_{\mathbf{Z},G}$ be arbitrary. The nearest neighbor interpolation in Example 1 motivates the assumption that, for a sampled function $\mathbf{u}$, there exists $\boldsymbol{\Phi}$ such that $\mathbf{u}^{(0)} = \mathbf{u} \circ \boldsymbol{\Phi}$. Recalling the concepts presented in Section 2 we consider the functional defined in (7) restricted to the set $\mathcal{U}_{\mathbf{Z},G}$ in order to reconstruct $\mathbf{u}$ from given $\mathbf{u}^{(0)}$. In turn, we restrict the flow equation (13) to $\mathcal{U}_{\mathbf{Z},G}$:

$$\partial_t \mathbf{u} = P_{\mathcal{U}_{0,G}} \left((\varepsilon I + \nabla \mathbf{u}^T \nabla \mathbf{u}) \nabla \cdot \left(\frac{\nabla \mathbf{u}}{|\nabla \mathbf{u}|} \right) \right), \tag{18}$$

where

$$P_{\mathcal{U}_{0,G}}(v) = v - \|G\|^{-2}_{L^2(R^2)} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \langle G_{i,j}, v \rangle_{L^2(\Omega)} G_{i,j}$$

is applied on each component separately. Note that the assumption $\mathbf{u}^{(0)} \in \mathcal{U}_{\mathbf{Z},G}$ together with $\partial_t \mathbf{u} \in \mathcal{U}_{0,G}$ asserts that the solution $\mathbf{u}(t), t \geq 0$ stays in $\mathcal{U}_{\mathbf{Z},G}$. At this point we remark that there is no analytical theory guaranteeing the well posedness of (18).

Since (18) comprises a projection, in order to solve (18) numerically a time-explicit scheme with sufficiently small step size Δt is required.

4 Numerical Results

We compare our method consisting in numerically solving (18) to two standard interpolation methods, namely nearest neighbor and cubic interpolation, as well as to established, sophisticated interpolation methods proposed by Tschumperlé & Deriche [21] and by Roussos & Maragos [19]. The method of Tschumperlé & Deriche is implemented in the GREYCstoration software (see http://cimg.sourceforge.net/greycstoration/), for the method of Roussos & Maragos, test results are available from the site http://cvsp.cs.ntua.gr/∼tassos/PDEinterp/ssvm07res/.

In our method, the kernel function has to be chosen appropriately. We use

$$G(x, y) := \frac{1}{\int_{[-\frac{1}{2},\frac{1}{2}]^2} g_\sigma(x, y)\, dx\, dy} \chi_{[-\frac{1}{2},\frac{1}{2}]^2}\, g_\sigma(x, y),$$

where g_σ is the two-dimensional isotropic Gaussian of standard deviation σ. In our method a value of 20 is used for the variance σ^2.

For evaluating the methods, we use the two test images shown in Fig. 1. For both images, a low and a high resolution version is available, where the low resolution image is obtained from the high resolution image via low-pass filtering (convolution with a bicubic spline) and downsampling by a factor of four, see [19]. The test images were provided by Roussos & Maragos.

The methods mentioned above are used to upsample the low resolution image by a factor four.

Our method is applied with 100 time steps, $\Delta t = 0.03$, $\varepsilon = 0.05$ and $\sigma^2 = 20$ for the first and 100 time steps, $\Delta t = 0.05$, $\varepsilon = 0.01$ and $\sigma^2 = 20$ for the

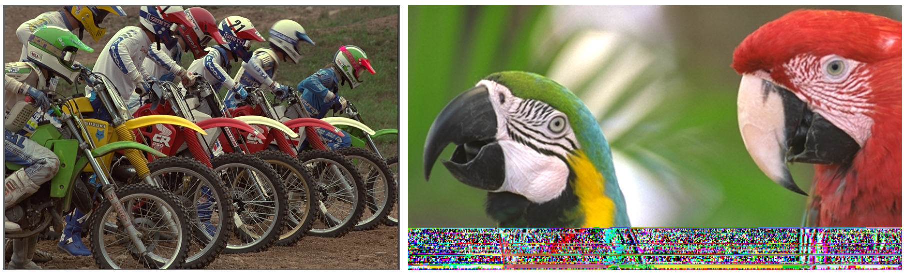

Fig. 1. Two test images. Each test image is available in a low and a high resolution version with a factor of four between both resolution.

second test image, respectively. For GREYCstoration (version 2.9) we use the option '-resize' together with the aimed size of the high resolution image and parameters '-anchor true', '-iter 3' and '-dt 10'. For the remaining parameters the default values are used. The results of Roussos' method were obtained from the web site mentioned above.

Let us consider the results of upsampling the first test image. In order to highlight the differences between the methods, we compare only details of the resulting images, see Fig. 2.

The results with nearest neighbor and cubic interpolation are shown in Fig. 2, top right and middle left, respectively. Both results are unsatisfactory and confirm, what is well known from the literature, that by nearest neighbor interpolation the upsampled images look blocky and cubic interpolation produces blurry images. The result of GREYCstoration with interpolation constraints (Fig. 2, middle row right) also appears blurry, but compared to cubic interpolation better reconstruct the edges in the image. The method proposed by Roussos & Maragos as well as our method (see Fig. 2, bottom row) produce sharp and well reconstructed edges.

In order to further investigate the differences between the PDE based methods, we zoom into two regions of the second test image, one region containing an edge (see Fig. 3) and one region with texture (see Fig. 4).

Fig. 3 shows the edge region after applying the methods proposed by Tschumperlé with interpolation constraints (top row, second left), Roussos (top row, second right) and our method (top row, right). For comparison we have plotted the detail of the original image (top row, left). One can see that by Tschumperlé's method the edges appear blurry and irregular. This seems to be an effect of the interpolation constraints, because when Tschumperlé's method is applied without constraints, strong anisotropic diffusion along the edge occurs so that the edge becomes more regular. By the method of Roussos the edge is reconstructed in a sharp way, but overshots appear. Our method is also able to reconstruct the edge sharply but with little overshots. Concerning the gray mark at the parrot's beak, we observe that Tschumperlé's method reconstructs the shape of the mark better than the other methods do.

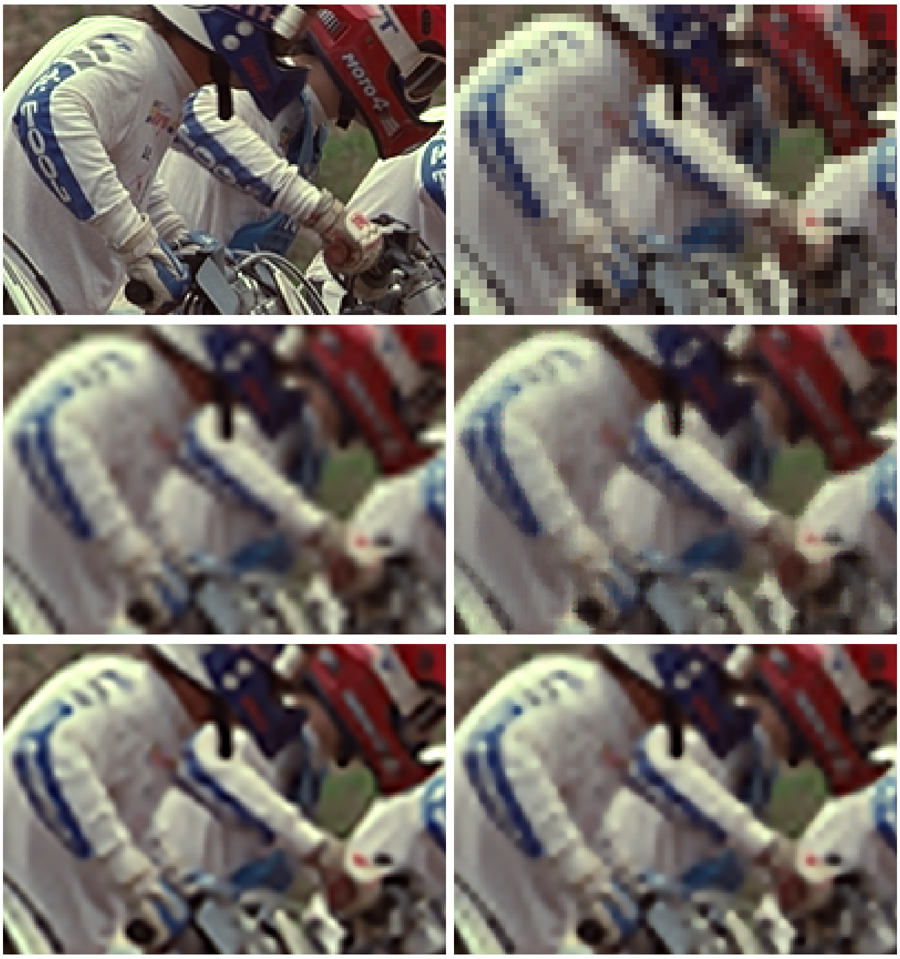

Fig. 2. Upsampling by a factor of four, Detail of the first test image. *top left:* original high resolution image, *top right:* nearest neighbor interpolation, *middle left:* cubic interpolation, *middle right:* interpolation using GREYCstoration, *bottom left:* interpolation method proposed by Roussos et. al, *bottom right:* proposed interpolation method

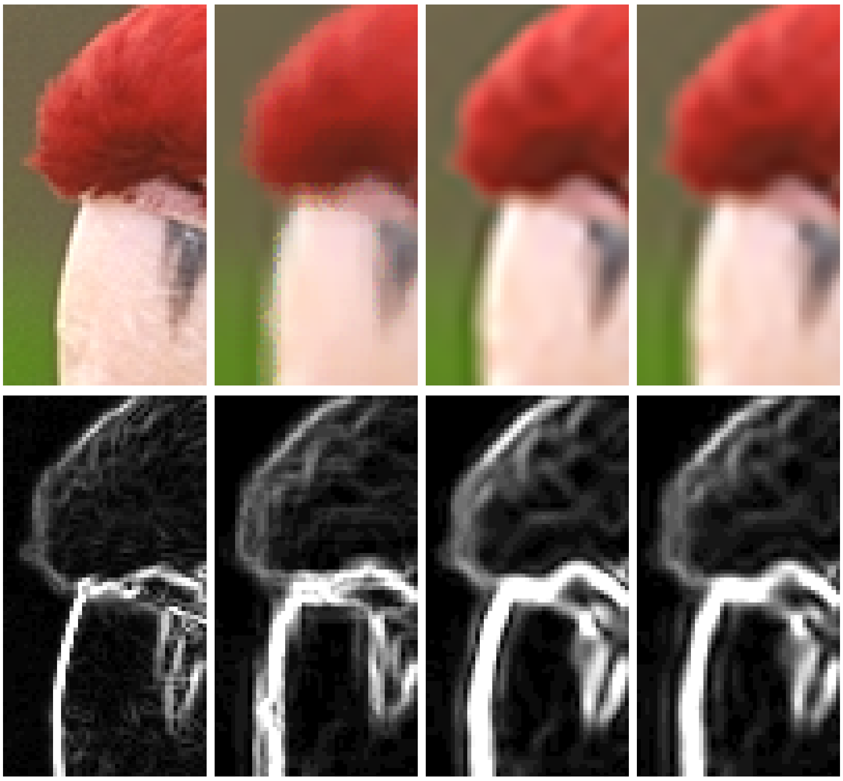

Fig. 3. Detail of an edge in the original and interpolated images (*top row*, using GREYCstoration with interpolation constraints, Roussos' method, and the proposed method) and subsequently applied Sobel operator (*bottom row*).

The differences in the behavior of the methods can also be recognized when applying the Sobel operator to the interpolated images: The thickness of the edges in the result of the Sobel operator indicates the blurriness of the reconstructed edge. We see that the proposed method produces sharper edges than the method by Roussos and more regular edges than the method by Tschumperlé. The overshots introduced by Roussos' method can also be observed in the outcome of the Sobel operator. They are far stronger than the overshots produced by our method.

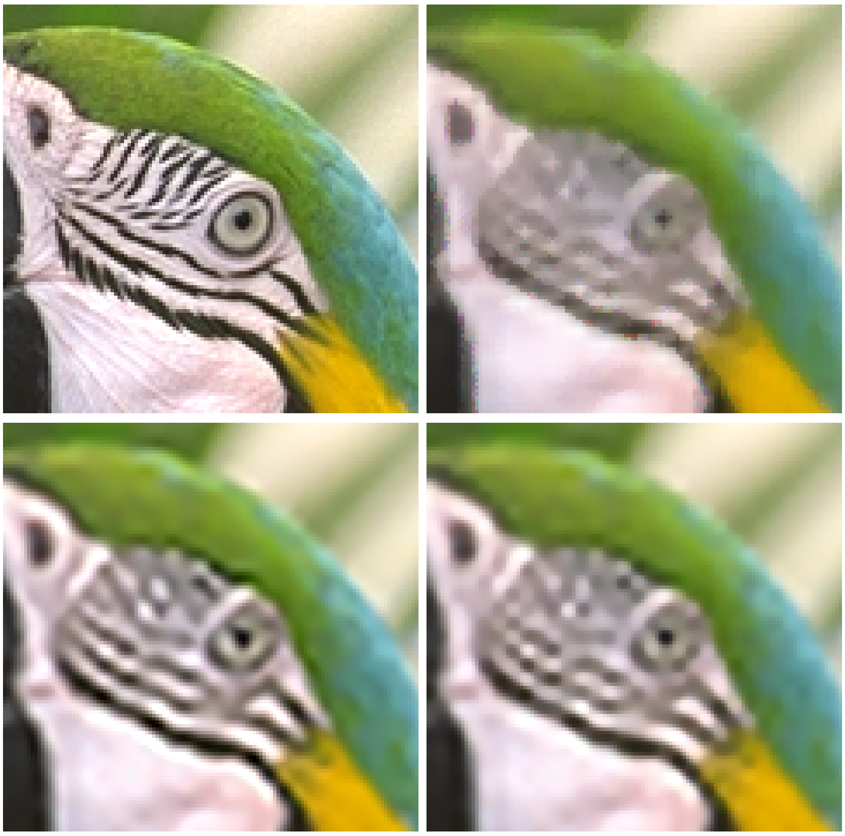

Fig. 4. A texture detail of the original (*top left*) and interpolated images using GREYCstoration (*top right*), Roussos' method (*bottom left*) and the proposed method (*bottom right*).

Now we investigate the effect of the interpolation methods on textures. Fig. 4, top left, shows a textured region of the original image. The results of the methods proposed by Tschumperlé (with interpolation constraints) and Roussos are given in Fig. 4, top right and bottom left, respectively. The result of the proposed method is shown in Fig. 4, bottom right. One observes a certain blurriness in the results by Tschumperlé's method. As for the result before, we point out that incorporating the interpolation constraints seems to have a strong effect on the result. When applying GREYCstoration without imposing constraints, the results are much more influenced by the anisotropic diffusion and the edges and the texture are accentuated. In the result of the interpolation method proposed by Roussos, we see a strong effect of the anisotropic diffusion on the texture, so that the result is more visually appealing than the other results. Nevertheless, a comparison with the original image shows that original and reconstructed texture differ significantly. In particular the orientations of the short stripes in the face of the parrot are different. Note that the anisotropic diffusion induced by the direction of the texture also affects the pupil of the parrot. On the result of our method we remark that the reconstruction of the texture is quite conservative, i.e., we stay near the initial guess. The blockyness is slightly reduced by the evolution process. Taking a look at the eye of the parrot, the relation of our method to Mean Curvature Flow can be observed: The pupil is reconstructed as a perfectly circular shape.

5 Conclusion

We have proposed a new PDE based method for the interpolation of color images. The method differs from other state-of-the-art methods by the underlying evolution process. We use a PDE which is a generalized Mean Curvature Flow, whereas other methods are based on anisotropic diffusion. Interpolation constraints are satisfied by projecting the evolution process onto an adequate function space.

Numerical tests show that our method is competitive to state-of-the-art interpolation methods. Due to the Mean Curvature Flow nature of the method, edges are well reconstructed. Textures are treated in a conservative manner.

Acknowledgments

We want to thank Gerhard Dziuk (Univ. Freiburg), Peter Elbau (RICAM, Linz) and Markus Grasmair (University Innsbruck) for inspirational discussions. We thank David Tschumperlé for providing GREYCstoration and Anastasios Roussos and Petros Maragos for providing the test images as well as the results of their algorithm.

The work of O.S. is partially funded by the project FSP S 92 (subproject 9203-N12).

References

1. A. Belahmidi and F. Guichard. A partial differential equation approach to image zoom. In *Proc. of the 2004 Int. Conf. on Image Processing*, pages 649–652, 2004.

2. M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *[13]*, pages 417–424, 2000.
3. H. Brézis. *Opérateurs maximaux monotones et semi-groupes de contractions dans les espaces de Hilbert*. North-Holland Publishing Co., Amsterdam, 1973. North-Holland Mathematics Studies, No. 5. Notas de Matemática (50).
4. W. Burger and M.J. Burge. *Digitale Bildverarbeitung*. Springer, 2005.
5. R. Chan, S. Setzer, and G. Steidl. Inpainting by flexible Haar wavelet shrinkage. Preprint, University of Mannheim, 2008.
6. T. Chan, S. Kang, and J. Shen. Euler's elastica and curvature based inpaintings. *SIAM J. Appl. Math.*, 63(2):564–592, 2002.
7. B. Dacorogna. *Weak Continuity and Weak Lower Semicontinuity of Non-Linear Functionals*, volume 922 of *Lecture Notes in Mathematics*. Springer Verlag, Berlin, Heidelberg, New York, 1982.
8. B. Dacorogna. *Direct Methods in the Calculus of Variations*, volume 78 of *Applied Mathematical Sciences*. Springer Verlag, Berlin, 1989.
9. P. Elbau, M. Grasmair, F. Lenzen, and O. Scherzer. Evolution by non-convex energy functionals. Reports of FSP S092 - "Industrial Geometry" 75, University of Innsbruck, Austria, 2008. Submitted.
10. M. Grasmair, F. Lenzen, A. Obereder, O. Scherzer, and M. Fuchs. A non-convex PDE scale space. In *[15]*, pages 303–315, 2005.
11. F. Guichard and F. Malgouyres. Total variation based interpolation. In *Proceedings of the European Signal Processing Conference*, volume 3, pages 1741–1744, 1998.
12. H. Hagen and J. Weickert, editors. *Visualization and Processing of Tensor Fields*. Mathematics and Visualization. Springer Verlag, 2006.
13. S. Hoffmeyer, editor. *Proceedings of the Computer Graphics Conference 2000 (SIGGRAPH-00)*. ACMPress, New York, 2000.
14. B. Jähne. *Digitale Bildverarbeitung*. Springer, 5th edition, 2002.
15. R. Kimmel, N. A. Sochen, and J. Weickert, editors. *Scale Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*. Springer, 2005.
16. F. Malgouyres and F. Guichard. Edge direction preserving image zooming: a mathematical and numerical analysis. *SIAM J. Numer. Anal.*, 39:1–37, 2001.
17. M.Z. Nashed, editor. *Generalized inverses and applications*. Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1976.
18. A. Roussos and P. Maragos. Vector-valued image interpolation by an anisotropic diffusion-projection pde. In *Scale Space and Variational Methods in Computer Vision*, volume 4485, pages 104–115. Springer, 2007.
19. A. Roussos and P. Maragos. Reversible interpolation of vectorial images by an anisotropic diffusion-projection pde. In *Special Issue for the SSVM '07 conference*. Springer, accepted for publication.
20. O. Scherzer and J. Weickert. Relations between regularization and diffusion filtering. *J. Math. Imaging Vision*, 12(1):43–63, 2000.
21. D. Tschumperlé. Fast anisotropic smoothing of multi-valued images using curvature-preserving pde's. *International Journal of Computer Vision (IJCV)*, 68:65–82, 2006.
22. D. Tschumperlé and R. Deriche. Vector valued image regularization with pdes: A common framework for different applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 2005.
23. J. Weickert and M. Welk. Tensor field interpolation with pdes. In *[12]*, pages 315–325, 2006.