

Infinite Dimensional Optimization Models and PDEs for Dejittering

Guozhi Dong¹, Aniello Raffaele Patrone¹, Otmar Scherzer^{1,2}, Ozan Öktem³

¹ Computational Science Center,
University of Vienna,
Oskar-Morgenstern-Platz 1, 1090 Wien, Austria
`{guozhi.dong,aniello.patrone,otmar.scherzer}@univie.ac.at`

² Johann Radon Institute for Computational and
Applied Mathematics (RICAM),
Austrian Academy of Sciences,
Altenberger Strasse 69, A-4040 Linz, Austria

³ Department of Mathematics,
KTH - Royal Institute of Technology,
Lindstedtsvägen 25, SE-10044 Stockholm, Sweden
`ozan@kth.se`

Abstract. In this paper we do a systematic investigation of continuous methods for pixel, line pixel and line dejittering. The basis for these investigations are the discrete line dejittering algorithm of Nikolova and the partial differential equation of Lenzen et al for pixel dejittering. To put these two different worlds in perspective we find infinite dimensional optimization algorithms linking to the finite dimensional optimization problems and formal flows associated with the infinite dimensional optimization problems. Two different kinds of optimization problems will be considered: Dejittering algorithms for determining the displacement and displacement error correction formulations, which correct the jittered image, without estimating the jitter. As a by-product we find novel variational methods for displacement error regularization and unify them into one family. The second novelty is a comprehensive comparison of the different models for different types of jitter, in terms of efficiency of reconstruction and numerical complexity.

Keywords: Dejittering, Variational methods, Nonlinear evolution PDEs

1 Introduction

A frequent task in image processing is *dejittering*, which is the process of assigning pixel positions to image data recorded with pixel displacements. Jitter is a type of distortions which arises frequently in signal processing, when the distance (time) between sampling points vary rendering signal errors. A specific form of jitter is line jitter that consists of horizontal shifts of each row (line) of an image. The shift is the same for the entire row. This may typically happen when

digitizing analog noisy video frames and there are line registration problems due to bad synchronization pulses. The effect is that the image lines are (randomly) shifted with respect to their original location, so vertical lines become jagged resulting in a disturbing visual effect since all shapes become jagged. One may also have line pixel jitter where pixels in a row are shifted differently. Finally there is pixel jitter where one also experiences vertical shifts.

The main goal of this paper is to establish relations between discrete and continuous models for dejittering. In particular we consider line, line pixel, and pixel jitter. In the literature these problems have been considered in an infinite dimensional continuous and in a finite dimensional discrete setting, resulting in different problem formulations and analysis. To link these approaches and put the theory on solid grounds (based on an infinite dimensional - discretization free - theory) we require to link the approaches.

Presently there exists two kind of algorithms for dejittering which we catalog as follows:

- *Dejittering algorithms* find the displacements by an optimization routine first and then restore the image by composing the jittered image with the displacement.
- *Displacement correction algorithms* compute the image directly without calculating the displacement function first.

The algorithms will be implemented for different purposes: For dejittering we assume a deterministic jitter, while in the later we assume a random perturbation.

Starting point of this paper are publications in different worlds, which deal with dejittering: The discrete optimization formulation of Nikolova [13,14] and Lenzen et al [8,9], which deals with displacement correction. We are generalizing Nikolova’s algorithm to the infinite dimensional setting and then establish a relation to displacement correction and systems of partial differential equations.

As a consequence we can discuss advantages and shortcuts of the different methods and discretization dependence.

The outline of this paper is as follows: In Section 2 we make the basic problem formulation for three types of jittering. Then we explain line dejittering and recall the standard formulation in the field from Nikolova [14] in Section 3. After deriving a continuous variant, we put this algorithm in perspective with displacement error regularization [3,8,9,15,16]. We explain the different philosophies but show the close relation of these areas in the general setting of line pixel dejittering; cf. Section 4. Moreover, we review continuous algorithms for pixel dejittering in Section 5. In Section 6 we formulate partial differential equations, which constitute the flows according to the continuous optimization energies. Finally we present numerical results in Section 7. The paper ends with a conclusion, where we outline the novelties of this work.

2 Basic Notation and Problem Formulation

In this paper we use the following notations:

- u can either denote a discrete (digital) gray valued image, in which case it is represented as a matrix $u \in \mathbb{R}^{m \times n}$, where m is number of columns, and n is number of rows, or
- u denotes a function $u : \Omega \rightarrow \mathbb{R}$ on the unit-square $\Omega = [0, 1]^2$. For a continuous image $u : \Omega \rightarrow \mathbb{R}$, one way to have the digitized image pixels is

$$u_{ij} = \frac{1}{h_x h_y} \int_{(i-1)/m}^{i/m} \int_{(j-1)/n}^{j/n} u(x, y) d(x, y).$$

Here, the pixel size is $h_x \times h_y$, with $h_x = \frac{1}{m}$ and $h_y = \frac{1}{n}$.

- η_{ij} and $\eta : \Omega \rightarrow \mathbb{R}$ denote noise. In the discrete setting the lines are horizontally numbered from bottom to top.

Let u^δ denote either a discrete, jittered image - then it is a matrix in $\mathbb{R}^{m \times n}$, or a continuous, jittered image, then it is function $u^\delta : \Omega \rightarrow \mathbb{R}$. Assuming that u denotes the original image without jittering, we consider the following discrete and continuous problem formulations:

Line jitter:

$$u^\delta(i, j) = u(i + \mathbf{d}_j, j) + \eta_{ij}, \quad u^\delta(x, y) = u(x + \mathbf{d}(y), y) + \eta(x, y), \quad (1)$$

respectively, where $\mathbf{d}_j \in \mathbb{Z}$ denotes the discrete jitter of the j -th line, and $\mathbf{d} : [0, 1] \rightarrow \mathbb{R}$ denotes the jitter function of the y -th component.

Line pixel jitter:

$$u^\delta(i, j) = u(i + \mathbf{d}_{i,j}, j) + \eta_{ij}, \quad u^\delta(x, y) = u((x + \mathbf{d}(x, y), y) + \eta(x, y), \quad (2)$$

respectively, where $\mathbf{d}_{i,j} \in \mathbb{Z}$ denotes the discrete jitter of the i -th pixel in the j -th line, and $\mathbf{d} : \Omega \rightarrow \mathbb{R}$ denotes the jitter function of the point (x, y) in x -direction.

Pixel jitter:

$$u^\delta(i, j) = u((i, j) + \mathbf{d}_{i,j}) + \eta_{ij}, \quad u^\delta(x, y) = u((x, y) + \mathbf{d}(x, y)) + \eta(x, y), \quad (3)$$

respectively, where $\mathbf{d}_{i,j} \in \mathbb{Z}^2$ denotes the discrete jitter of the (i, j) -th pixel, and $\mathbf{d} : \Omega \rightarrow \mathbb{R}^2$ denotes the jitter vector field at the point (x, y) .

For those jittered pixels which run out of the domain of the original image u , we define their intensity values as 0.

In the literature, many dejittering algorithms are particularly designed for line jittering, referring to (1), see for instance [6, 7, 13, 14, 18]. In these algorithms, the jittering error is considered deterministic, and a probably noisy input image has to be smoothed in an additional step, either before or after dejittering. The problems of line pixel jitter (2) and pixel jitter (3) have been discussed for instance in [8, 9], where a displacement error correction model has been considered. In this context, it is commonly assumed that noise is significant and jitter is stochastic, and the methods are supposed to dejitter and denoise simultaneously.

3 Line Dejittering

In this section we investigate algorithms for line dejittering. After reviewing algorithms from the literature, we will formulate line pixel and pixel dejittering below.

As we have mentioned in the introduction, there are two different kinds of algorithms for dejittering in the literature. The prime example of the first type approach is Nikolova's algorithm [13,14], which is outlined below. A-priori Nikolova's approach is formulated in a discrete setting. We provide a continuous formulation below, which allows us to put it in perspective with the second approach, and thus in turn to partial differential equation models in the spirit of [8,9].

3.1 Nikolova's Algorithm for Discrete Line Dejittering

Nikolova [13,14] proposed an efficient algorithm for discrete line dejittering. This algorithm is based on energy minimization and determines in an iterative way, from bottom to top, for each horizontal image line discrete integer values $\mathbf{d}_j, j \in \{1, 2, \dots, n\}$, which indicate the horizontal displacement of the j -th line, respectively.

The algorithm involves setting values of an exponential parameter p , which Nikolova chooses as $p = 1$ or $p = 0.5$, $p = 0.5$ is better suited for discontinuous images, while $p = 1$ is better suited for smooth images. Moreover, it is assumed that the jitter is bounded, such that there is a parameter σ constraining the maximal line jitter (a typical values is $\sigma = 6$ pixels):

$$|\mathbf{d}_j| \leq \sigma, \quad \forall j = 2, \dots, n.$$

1. The algorithms is initialized by setting $j := 2$, $\mathbf{d}_1 := 0$, $\hat{u}(i, 1) := u^\delta(i, 1)$ and selecting the parameter $\sigma^* \geq \sigma$. The minimizer $\hat{\mathbf{d}}_2$ of the functional

$$\mathcal{J}_2(\mathbf{d}_2) := \sum_{i=\sigma^*+1}^{m-\sigma^*} |u^\delta(i - \mathbf{d}_2, 2) - u^\delta(i, 1)|^p \quad (4)$$

is used to define $\hat{u}(i, 2) := u^\delta(i - \hat{\mathbf{d}}_2, 2)$.

2. For $j = 3, \dots, n$ determine $\hat{\mathbf{d}}_j$ as the minimizer of the functional

$$\mathcal{J}_j(\mathbf{d}_j) := \sum_{i=\sigma^*+1}^{m-\sigma^*} |u^\delta(i - \mathbf{d}_j, j) - 2\hat{u}(i, j-1) + \hat{u}(i, j-2)|^p, \quad (5)$$

and define $\hat{u}(i, j) = u^\delta(i - \hat{\mathbf{d}}_j, j)$.

3.2 A Continuous Optimization Problem for Line Dejittering

We here formulate a continuous variant of Nikolova's algorithm, which also establishes the relation to existing variational methods and partial differential

equations for dejittering. Let $u^\delta : \Omega \rightarrow \mathbb{R}$ be the line jittered variant of u , so u^δ satisfies (1). In order to recover u and \mathbf{d} , we minimize (6) for each $\hat{y} \in [0, 1]$ separately, where \hat{y} indicates the continuum position of the line in the image,

$$\mathcal{J}_c(\mathbf{d})(\hat{y}) := \lim_{\tau \rightarrow 0^+} \frac{1}{2\tau} \int_{\max\{\hat{y}-\tau, 0\}}^{\min\{\hat{y}+\tau, 1\}} \int_{\sigma_*}^{1-\sigma_*} |\partial_y^k u^\delta(x - \mathbf{d}(y), y)|^p d(x, y), \quad (6)$$

subject to

$$\|\mathbf{d}\|_{L^\infty([0,1])} \leq \sigma. \quad (7)$$

The parameter σ_* is chosen to satisfy $\sigma \leq \sigma_*$. With this choice the integrand in the integral $\int_{\sigma_*}^{1-\sigma_*} |\partial_y^k u^\delta(x - \mathbf{d}(y), y)|^p dx$ is evaluated only for arguments of u^δ in the interior of the image domain $[0, 1] \times [0, 1]$. This correspond to the discrete sum $\sum_{i=\sigma_*+1}^{m-\sigma_*}$ in the Nikolova algorithm. The term $\partial_y^k u^\delta$ denotes the k -th derivative of u^δ with respect to the second component. Since

$$\frac{u^\delta(i - \mathbf{d}_j, j) - 2\hat{u}(i, j-1) + \hat{u}(i, j-2)}{h_y^2} \approx \partial_y^2 u^\delta(ih_x - \mathbf{d}((j-1)h_y), (j-1)h_y),$$

we propose the following simplified variant of (6) and (7), namely to minimize

$$\mathcal{J}^{(k)}(\mathbf{d}) := \frac{1}{p} \int_{\Omega} |\partial_y^k u^\delta(x - \mathbf{d}(y), y)|^p d(x, y) \quad (8)$$

subject to

$$\|\mathbf{d}\|_{L^2([0,1])} \leq \hat{\sigma}. \quad (9)$$

The main difference to minimizing \mathcal{J}_c is that we consider integration over all of Ω . To make this well-defined, we propose to extend u^δ symmetric across left and right, and top and bottom images boundaries, respectively. Another difference is that we consider an *a joint* approach, which optimizes globally over all pixels, instead of separately for each line. Moreover, from a modelling point of view taking the second derivative ($k = 2$) of u^δ in the functional \mathcal{J}_c is not mandatory, for instance, we may take as well the derivative ($k = 1$) or another integer order. In practice, minimizing the functional with second order derivatives performs better than using first order derivatives in a noise free environment. For the other parameter p in (8), in the discrete setting, Nikolova has suggested to use either 0.5 or 1, however, we would propose to choose either $p = 1$ or $p = 2$, in order to keep the convexity of the functional in our continuous model, where $p = 1$ works better with the discontinuities.

4 Line Pixel Dejittering

In this section we review line pixel dejittering and displacement regularization: We find that within the continuous setting, formally, the optimization approach for line dejittering from last section can be similarly generalized to the case of line

pixel dejittering. However, the formal difference is that for line pixel dejittering $\mathbf{d} : \Omega \rightarrow \mathbb{R}$ is a bounded random field over the whole two dimensional domain Ω , while for line jitter $\mathbf{d} : [0, 1] \rightarrow \mathbb{R}$. Thus, we propose to optimize the functional which is only slightly changed from (8)

$$\mathcal{J}_2^{(k)}(\mathbf{d}) := \frac{1}{p} \int_{\Omega} |\partial_y^k u^\delta(x - \mathbf{d}(x, y), y)|^p d(x, y) \quad (10)$$

subject to $\|\mathbf{d}\|_{L^2(\Omega)} \leq \hat{\sigma}$.

Because we assume small displacements \mathbf{d} , we also consider approximating the term $\partial_y^k u^\delta(x - \mathbf{d}(x, y), y)$ by its linearisation:

$$\partial_y^k u^\delta(x - \mathbf{d}(x, y), y) \approx \partial_y^k u^\delta(x, y) - \mathbf{d}(x, y) \partial_x \partial_y^k u^\delta(x, y).$$

Replacing the nonlinear term by its linearization, we arrive at the constrained optimization problem, which is to minimize

$$\mathcal{J}_2^{(k)}(\mathbf{d}) := \frac{1}{p} \int_{\Omega} |\partial_y^k u^\delta(x, y) - \mathbf{d}(x, y) \partial_x \partial_y^k u^\delta(x, y)|^p d(x, y), \quad k = 1, 2 \quad (11)$$

subject to (9).

For $1 < p \leq 2$, $\mathcal{J}_2^{(k)}$ is strictly convex, and for three-times continuously differentiable u^δ also weakly lower semi-continuous. Then, the constrained optimization problem is equivalent to the method of Tikhonov-regularization with parameter choice by Morozov's discrepancy principle, consisting in calculation of

$$\mathbf{d}(\alpha) := \arg \min_{\mathbf{d}} \left\{ \mathcal{J}_2^{(k)}(\mathbf{d}) + \frac{\alpha}{2} \|\mathbf{d}\|_{L^2(\Omega)}^2 \right\}, \quad (12)$$

where α is chosen to satisfy $\|\mathbf{d}(\alpha)\|_{L^2(\Omega)} = \hat{\sigma}$. For further background on the relation between Tikhonov regularization and constrained optimization problems see for instance [1,4,5,10,11,17,19,20]. For $p \leq 1$ the relation is not obvious, but we ignore this difficulty.

We stress the fact that the minimizer of (12) with $p = 2$ can be explicitly calculated: We have

$$\mathbf{d}(\alpha) = \frac{\partial_y^k u^\delta \partial_x \partial_y^k u^\delta}{\alpha + (\partial_x \partial_y^k u^\delta)^2}. \quad (13)$$

This explicit linearised method provides insufficient results (cf. Figure 1).

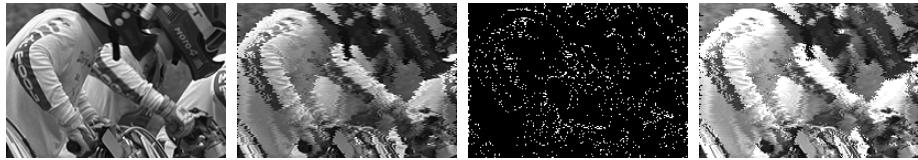


Fig. 1. Left to right: ground truth, line jittered image, displacement, recovered image

4.1 Displacement Error Correction for Line Pixel Dejittering

In the following we outline an approach for dejittering, which does not recover the jitter but the dejittered image directly. We use a first order approximation of the data by assuming that the jitter is only a small disturbance:

$$u^\delta(x, y) \approx u(x + d(x, y), y) \approx u(x, y) + \partial_x u(x, y)d(x, y). \quad (14)$$

Considering the approximation as an identity we find that

$$d(x, y) = \frac{u^\delta(x, y) - u(x, y)}{\partial_x u(x, y)}. \quad (15)$$

Now, instead of minimizing $\mathcal{J}_2^{(k)}$ with respect to d , we replace in $\mathcal{J}^{(k)}$ the u^δ by $u(x + d(x, y))$ and use the identity (15), and minimize with respect to u . Thus the optimization problem for line pixel dejittering consists in the minimization of the functional:

$$\mathcal{N}(u) := \alpha \frac{1}{2} \int_{\Omega} \left| \frac{u^\delta(x, y) - u(x, y)}{\partial_x u(x, y)} \right|^2 d(x, y) + \underbrace{\frac{1}{p} \int_{\Omega} |\partial_y^k u(x, y)|^p d(x, y)}_{\mathcal{R}}. \quad (16)$$

Remark 1. When we use this approach to correct for line jitter, we have to respect the fact that each line has the same shift, which leads to

$$0 = \partial_x d(y) \approx \partial_x \left(\frac{u^\delta(x, y) - u(x, y)}{\partial_x u(x, y)} \right).$$

Thus line jitter correction can be rephrased as an unconstrained minimization of the functional

$$\mathcal{N}(u) + \beta \int_{\Omega} \left(\partial_x \left(\frac{u^\delta(x, y) - u(x, y)}{\partial_x u(x, y)} \right) \right)^2 d(x, y), \quad (17)$$

where β is a penalty parameter.

5 Pixel Dejittering

The problem of pixel jitter correction can be formulated again as a constraint optimization problem, consisting in minimization of

$$\mathcal{J}_3^{(k)}(d) := \frac{1}{p} \int_{\Omega} |\partial_y^k u^\delta((x, y) - d(x, y))|^p d(x, y) \quad (18)$$

subject to $\|d\|_{(L^2(\Omega))^2} \leq \hat{\sigma}$. Note the fundamental difference that $d : \Omega \rightarrow \mathbb{R}^2$, while for line pixel jitter $d : \Omega \rightarrow \mathbb{R}$, and for line jitter $d : [0, 1] \rightarrow \mathbb{R}$.

Displacement error regularization for correcting pixel jitter has been considered in [8,9]. It is again based on Taylor expansion

$$u^\delta(x, y) - u(x, y) \approx d \cdot \nabla u,$$

which implies that we can choose as a solution $\mathbf{d} \approx (\nabla u)^\dagger(u^\delta - u)$, where $(\nabla u)^\dagger$ denotes the Moore-Penrose pseudo-inverse of ∇u . This choice of an inverse of ∇u considers displacement errors which are orthogonal to level lines of u .

Here, we define

$$\hat{\mathcal{S}}(u) := \frac{1}{2} \|(\nabla u)^\dagger(u^\delta - u)\|_{L^2(\Omega)}^2.$$

Assuming that u is of finite total variation we ended up with the following regularization functional [8,9]:

$$\hat{\mathcal{N}}(u) := \alpha \hat{\mathcal{S}}(u) + \int_{\Omega} |\nabla u(x, y)| d(x, y). \quad (19)$$

Note that in comparison with (16), $\int_{\Omega} |\partial_y^k u(x, y)|^p d(x, y)$ has been replaced by the TV -semi norm $\int_{\Omega} |\nabla u(x, y)| d(x, y)$.

6 PDE Models as Formal Energy Flows

Considering \mathcal{S} as a metric, the minimization of functional \mathcal{N} defined in (16), can be formally solved as metric flows of \mathcal{S} with energy \mathcal{R} . In [9], a PDE according to (19) has been derived by considering $\hat{\mathcal{N}}(\alpha, \cdot)$ as an implicit time-step of the associated flow, following that, we state the flows according to (16) and (19).

– The flow associated with (16), for $k = 1, 2$ and $p = 1, 2$ is:

$$\begin{cases} \partial_t u = |\partial_x u|^2 \partial_y^k \left(\frac{\partial_y^k u}{|\partial_y^k u|^{2-p}} \right); \\ u = u^\delta, \quad \text{in } \Omega \times \{0\}; \\ \partial_y^{2l-1} u = 0, \quad \text{on } \{0, 1\} \times [0, 1], \quad \forall l = 1, \dots, k. \end{cases} \quad (20)$$

– We emphasize that the flow associated to (19) is

$$\begin{cases} \partial_t u = |\nabla u|^2 \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right); \\ u = u^\delta, \quad \text{in } \Omega \times \{0\}; \\ \partial_n u = 0, \quad \text{on } \partial\Omega. \end{cases} \quad (21)$$

7 Numerical Results

In this section we show the numerical results of our newly developed model (20) for different choices of k and p , making comparisons with the approach from [9], that consists in solving (21), and with Nikolova's algorithm [14]. In the implementation, for $p = 2$ in (20), we use standard finite differences discretization with semi-implicit iteration, but for the case of $p = 1$, the solution of (20) is

obtained by solving the convex optimization problem (22) iteratively, where we generalised the *TV* denoising algorithm from [2] to approximate the solution.

$$\begin{cases} u^{m+1} := \arg \min_u \left\{ \frac{\alpha}{2} \int_{\Omega} \frac{|u^m(x, y) - u(x, y)|^2}{|\partial_x u^m(x, y)|^2 + \epsilon} + |\partial_y^k u(x, y)| d(x, y) \right\}, \\ u^0 = u^\delta. \end{cases} \quad (22)$$

Here α corresponds to the time-stepping and $u^m \approx u(m\alpha)$. In all the experiments, we use as stopping criteria some threshold of $\|u^m - u^{m+1}\|_{L^2}$. The test data are generated by adding jitter to clean test images. In addition noisy test data are generated by composing the test image with Gaussian noise of mean 0 and standard deviation 10. In order to evaluate the results quantitatively, we consider the *mean square error* (MSE) computed by averaging the intensity difference between the analyzed pixel $\hat{u}(i, j)$ and the reference pixel $u(i, j)$, and the related quantity of *peak signal to noise ration* (PSNR)

$$MSE = \frac{1}{N} \sum_{i=1}^m \sum_{j=1}^n (\hat{u}(i, j) - u(i, j))^2 \quad \text{and} \quad PSNR = 10 \log_{10} \frac{L^2}{MSE},$$

where L is the dynamic range of allowable pixel intensities, e.g. for an 8-bit per pixel image $L = 2^8 - 1 = 255$. These quantity are appealing but not well matched to perceived visual quality as reported in [12] and [21]. For that reason we consider also the *structural similarity* (SSIM) index [21] defined as:

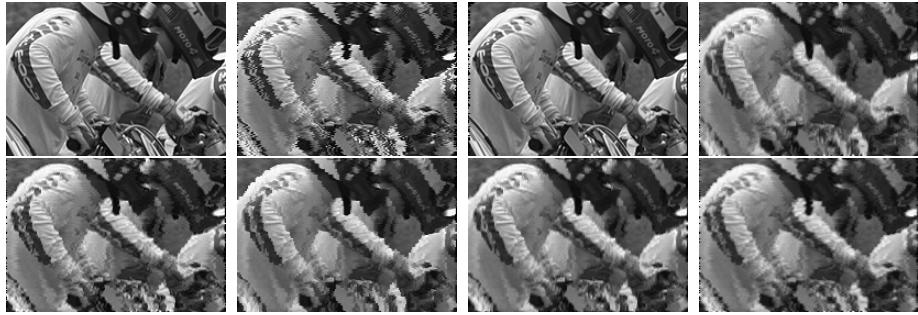
$$SSIM(\hat{u}, u) = f(l(\hat{u}, u), c(\hat{u}, u), s(\hat{u}, u)),$$

where the three independent components $l(\hat{u}, u)$, $c(\hat{u}, u)$, $s(\hat{u}, u)$ are the similarity functions of the luminance, the contrast and the structure, respectively, between the reconstructed and test image, and f is a combination function.

Quantitatively, the higher *PSNR* value the better similarity between the test data and the original clean image. Moreover, a small value of *MSE* points out a good intensity approximation of the original data, and a larger value of *SSIM* claims that the structure of the original image is better preserved.

Table 1 gives a comprehensive evaluation of different methods for image dejittering, which are the algorithm for solving (20) presented in this paper, and the algorithms from [9] and from [14], respectively. For the test images used for line dejittering and line pixel dejittering, we have not superimposed the data with additive noise. The test data used for pixel dejittered was considered with additive noise. For line dejittering, Nikolova's algorithm [14] gives the most superior results. Evaluating the two different PDE models, we notice that (20) performs better than [9] for line dejittering. [14] is not able to handle line pixel dejittering, in contrast with the PDE models. In this case the method in [9] achieves slightly better grades than (20); see Table 1. However visually, one may find that (20) (e.g. with parameter $k = 2, p = 1$) has less blurring of the reconstructed image and keeps more clear details; see Fig 3. The highlight of the approach [9] happens in the pixel dejittering task, where it outperforms the

Measure	Test data	$k=1, p=2$	$k=2, p=2$	$k=1, p=1$	$k=2, p=1$	cf.[9]	cf.[14]
Line Jitter Data without Adding Noise							
PSNR	17.814	19.886	20.031	20.109	20.461	19.807	24.818
MSE	1075.7	667.407	645.584	634.035	584.668	679.740	214.408
SSIM	0.622	0.704	0.714	0.709	0.729	0.691	0.998
Line Pixel Jitter Data without Adding Noise							
PSNR	16.608	17.913	17.956	18.193	18.356	19.213	13.999
MSE	1420.0	1051.4	1040.9	985.634	949.517	779.525	2589
SSIM	0.484	0.552	0.558	0.566	0.571	0.618	0.308
Pixel Jitter Data with Adding Noise							
PSNR	15.367	17.460	17.563	17.688	17.891	19.064	-
MSE	1889.8	1167.1	1139.6	1137	1056.6	806.614	-
SSIM	0.316	0.433	0.461	0.457	0.487	0.585	-

Table 1. Comparison of noisy and noise free data affected by different jitter types.**Fig. 2.** Line Dejittering. Top row: The ground truth, the noisy free line jittered image, dejittered with [14], dejittered with (20) $k = 1, p = 2$. Bottom row: dejittered with (20) $k = 2, p = 2$, (20) $k = 1, p = 1$, (20) $k = 2, p = 1$, approach from [9].

others both quantitatively and qualitatively. Over all the tests, it is not hard to find that, for the model (20), the choice of parameter $k = 2, p = 1$ gives the most competitive results in compare with the other parameter choices.

8 Conclusion

The novelties of this paper are that we have shown the formal connection of Nikolova's method with variational displacement error correction and PDE methods. To do this, we have unified a family of variational methods for displacement error regularization, which apply for different dejittering applications. The second novelty is a comparison of the different models for different types of jitter. An analysis of the proposed algorithms for minimizing models (16) is lacking and this might be a future research topic. Another aspect will be to investigate problems in tomography, which involve reconstruction of objects that show small (unknown) displacements while being imaged.

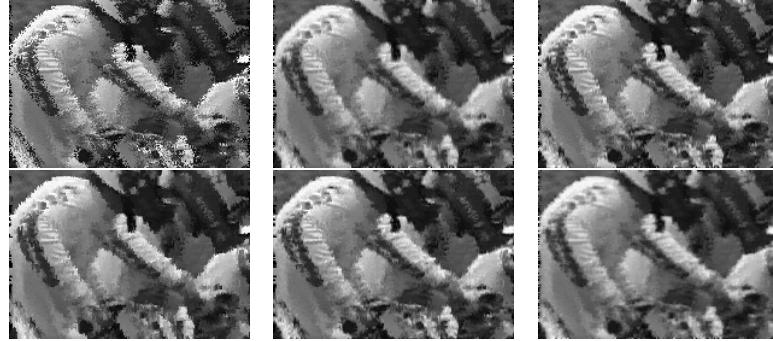


Fig. 3. Line Pixel Dejittering. Top row: The noisy line pixel jittered image, dejittered with (20) $k = 1, p = 2$, (20) $k = 2, p = 2$. Bottom row: (20) $k = 1, p = 1$, (20) $k = 2, p = 1$, approach from [9].

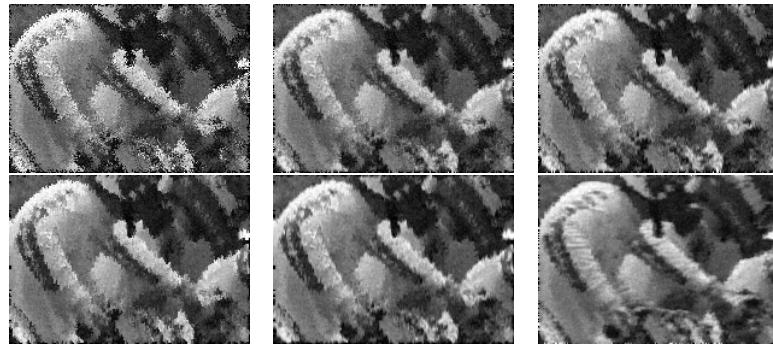


Fig. 4. Pixel Dejittering. Top row: The noisy line pixel jittered image, dejittered with (20) $k = 1, p = 2$, (20) $k = 2, p = 2$. Bottom row: (20) $k = 1, p = 1$, (20) $k = 2, p = 1$, approach from [9].

Acknowledgements

The work of GD and OS has been supported by the Austrian Science Fund (FWF) within the project *Variational Methods for Imaging on Manifolds* within the NFN Geometry and Simulation, project S11704. AP and OS are supported by the project *Modeling Visual Attention as a Key Factor in Visual Recognition and Quality of Experience* funded by the Wiener Wissenschafts und Technologie Funds - WWTF. The authors thank the reviewers for their comments to improve the presentation of the paper.

References

1. H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*, volume 375 of *Mathematics and its Applications*. Kluwer Academic Publishers Group, Dordrecht, 1996.

2. M.A.T. Figueiredo, J.B. Dias, and J. P. Oliveira et. al. On total variation denoising: A new majorization-minimization algorithm and an experimental comparison with waavlet denoising. In *IEEE International Conference on Image Processing*,, pages 2633–2636. IEEE, 2006.
3. M. Grasmair, F. Lenzen, A. Obereder, O. Scherzer, and M. Fuchs. A non-convex PDE scale space. In R. Kimmel, N. Sochen, and J. Weickert, editors, *SSVM*, volume 3459 of *LNCS*, pages 303–315. Springer, Berlin / Heidelberg, 2005.
4. C. W. Groetsch. *The Theory of Tikhonov Regularization for Fredholm Equations of the First Kind*. Pitman, Boston, 1984.
5. C. W. Groetsch. *Inverse Problems*. Mathematical Association of America, Washington, DC, 1999. Activities for undergraduates.
6. S. H. Kang and J. Shen. Video dejittering by bake and shake. *Image Vision Comput.*, 24(2):143–152, 2006.
7. S. H. Kang and J. Shen. Image dejittering based on slicing moments. In X.C. Tai, K.A. Lie, T. F. Chan, and S. Osher, editors, *Image Processing Based on Partial Differential Equations*. Springer, New York, 2007. Mathematics and Visualization.
8. F. Lenzen and O. Scherzer. A geometric pde for interpolation of m -channel data. In X.C. Tai, M. Knut, L. Marius, and L. Knut-Andreas, editors, *SSVM '09, LNCS*, pages 413–425, Berlin, Heidelberg, 2009. Springer-Verlag.
9. F. Lenzen and O. Scherzer. Partial differential equations for zooming, deinterlacing and dejittering. *Int. J. Comput. Vision*, 92(2):162–176, 2011.
10. V. A. Morozov. *Methods for Solving Incorrectly Posed Problems*. Springer Verlag, New York, Berlin, Heidelberg, 1984.
11. V. A. Morozov. *Regularization Methods for Ill-Posed Problems*. CRC Press, Boca Raton, 1993.
12. P. Ndjahah, H. Kikuchi, M. Yukawa, H. Watanabe et. al. An investigation on the quality of denoised images. *Int. J. Circ. Syst. Sign. Proc.*, 5:423–434, 2011.
13. M. Nikolova. Fast dejittering for digital video frames. In X.C. Tai, M. Knut, L. Marius, and L. Knut-Andreas, editors, *SSVM'09, LNCS*, pages 439–451. Springer Berlin Heidelberg, 2009.
14. M. Nikolova. One-iteration dejittering of digital video images. *J. Vis. Commun. Image Represent.*, 20:254–274, 2009.
15. O. Scherzer. Explicit versus implicit relative error regularization on the space of functions of bounded variation. In M. Z. Nashed and O. Scherzer, editors, *Inverse problems, image analysis, and medical imaging*, volume 313 of *Contemporary Mathematics*, pages 171–198. AMC, Providence, Rhode Island, 2002.
16. O. Scherzer. Scale space methods for denoising and inverse problem. *Adv. Imaging Electron Phys.*, 128:445–530, 2003.
17. O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen. *Variational methods in imaging*, volume 167 of *Applied Mathematical Sciences*. Springer, New York, 2009.
18. J. Shen. Bayesian video dejittering by bv image model. *SIAM J. Appl. Math.*, 64(5):1691–1708, 2004.
19. A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems*. John Wiley & Sons, Washington, D.C., 1977.
20. A. N. Tikhonov, A. Goncharsky, V. Stepanov, and A. Yagola. *Numerical Methods for the Solution of Ill-Posed Problems*. Kluwer, Dordrecht, 1995.
21. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.