A FAST DEJITTERING APPROACH FOR LINE SCANNING MICROSCOPY

Landry Duguet † *Julien Calve* ⋆ *Cyril Cauchois* ⋆ *Pierre Weiss* $^{\pi}$

† INRIA Lyon, RDP - ENS Lyon
⋆ INNOPSYS, Carbonne, France
$^{\pi}$ Institut de Mathématiques de Toulouse, CNRS, Université Paul Sabatier

ABSTRACT

We propose two efficient optimization approaches to correct jitter effects appearing in a specific type of line scanning microscopy. In this modality, even lines suffer from a non uniform and non integer distortion with respect to odd lines, creating significant visual artifacts. The huge image size make this problem highly challenging. To handle it, we propose two techniques. One is based on dynamic programming and has a complexity linear w.r.t. the number of pixels. The second is based on a convex relaxation and can be particularly efficient for parallel architectures. Both algorithms provide globally optimal solutions. The empirical reconstruction results are of high quality.

***Index Terms*—** dejittering, line scanning microscopy, dynamic programming, registration, convex relaxations.

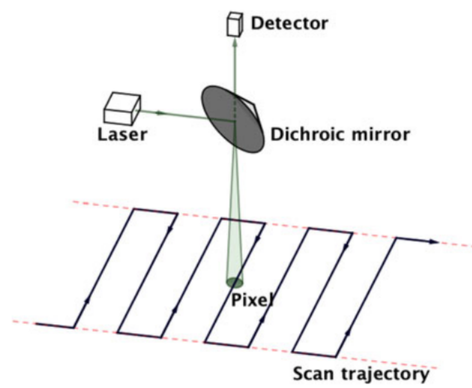


Fig. 1. Principle of a line scanning microscope

1. INTRODUCTION

Line scanning microscopy [1, 2, 3] is a technology that allows taking high-resolution and large-scale images of various biological samples in short times. For instance the InnoScan or InnoQuant systems `https://www.innopsys.com/microarray-scanners/` yield 2D images with a maximal pixel resolution of $0.2\mu m$ and of size up to 125000×375000 within 3 hours. Images are acquired pixel by pixel, with a change of scan direction between subsequent lines, see Fig. 1. This principle gives rise to jittering effects: shifts between the pixels in even and odd lines, see Fig. 3 (a) for an example of raw image taken with the InnoQuant system. This significantly impacts image interpretation, be it by human inspection or automatized algorithms.

A brief review of existing approaches Image dejittering is a problem that began receiving some attention for video artifacts correction in the 1990's [4, 5]. Therein the author proposed estimating the parameters of an autoregressive model. Later, [6, 7] proposed to minimize the ℓ^p-norm of the difference between consecutive lines with respect to an integer shift. This principle was later extended to off-the-grid shifts [8]. More recently, L. Lang proposed dynamic programming (DP) approaches to solve similar problems more efficiently with integer shifts [9]. Our DP approach is very similar in spirit with an adaptation to our particular setting. All those works consider models of jitter not accounting for the specificity of line scanning microscopes:

- The displacements live off-the-grid, making the estimation significantly more involved.
- The displacements vary along a line, while the initial models [4] considered constant shifts.
- The displacements are identical for even and odd lines, simplifying the estimation considerably.
- The images can be huge, making it necessary to derive linear time algorithms.

To the best of our knowledge, the only reference dealing with all the above specificities is a recent paper by Nguyen et al. [10], which will be described later but suffers from significant limitations.

Our contribution In this work, we consider that the even lines have not been shifted and can be used as a reference for the registration. We only seek to estimate the shifts for the odd lines. We propose two different variational approaches to solve this problem. Both of them provide globally optimal solutions contrarily to [10]. They provide satisfactory results in short computing times, compatible with the huge dimensions of the InnoQuant system. Dejittering result are shown Fig. 3.

2. PROBLEM DESCRIPTION

Acquisition model Let $f : \mathbb{R}^2 \to \mathbb{R}$ denote the continuous image we would like to acquire. We assume that it is smooth, which is a reasonable assumption, since light diffraction has a regularizing effect. We assume that the discrete sampled image reads:

$$\mathbf{g}[i,j] = \begin{cases} f(i,j) & \text{if } i \text{ is odd} \\ f(i, j+d_j) & \text{if } i \text{ is even,} \end{cases} \tag{1}$$

where $i \in [\![1,m]\!]$, $j \in [\![1,n]\!]$ and $d \in \mathbb{R}^n$ is the unknown displacement vector we would like to estimate. The number of pixels in the vertical and horizontal directions are $m \in 2\mathbb{N}$ and $n \in \mathbb{N}$. We let $\mathbf{u} = [\mathbf{u}_1, \ldots, \mathbf{u}_n] \in \mathbb{R}^{m/2\times n}$ and $\mathbf{v} = [\mathbf{v}_1, \ldots, \mathbf{v}_n] \in \mathbb{R}^{m/2\times n}$ denote the images composed of the odd and even lines with

$$\mathbf{u}_j = \mathbf{g}[1:2:m-1, j] \text{ and } \mathbf{v}_j = \mathbf{g}[2:2:m, j].$$

Displacement function An important property of line scanning microscopes is that the vector d is the discretization of a smooth function: it is related to mechanical or optical displacements, which are constrained by the physics. Another important feature is that the displacements are bounded. Throughout the paper, we will assume that $\mathbf{d} \in [-R, R]^n$, where $R \in \mathbb{N}$ is the maximal possible displacement.

Interpolation model Since f is assumed to be smooth, its values at non integer coordinates can be approximated from the samples on the grid. Throughout the paper, we will assume that

$$f(i,x) = \sum_{j=1}^{n} f(i,j)\psi(j-x) \text{ with } \psi(x) = (1-|x|)_+ \tag{2}$$

is the first order cardinal spline. For all $x \in [1,n]$, we let $u(\cdot, x) \in \mathbb{R}^m$ denote the continuous interpolant of $\mathbf{u}$. Alternative choices of interpolant with compact support (e.g. B-splines) could be considered, but we will stick to piecewise linear interpolation for simplicity.

3. RESOLUTION METHODS

In this section, we first review the approach proposed in [10] and then introduce our contributions.

3.1. The existing methodology

In [10], the authors propose to recover the displacement d by minimizing an energy inspired by the field of image registration. They suggest to solve:

$$\inf_{\mathbf{d}\in\mathbb{R}^n} G(\mathbf{d}) + \sum_{j=1}^{n} F_j(d_j) \text{ for } F_j(d_j) = \lambda \|u(\cdot, j+d_j) - \mathbf{v}_j\|_q^q \tag{3}$$

where $\|\cdot\|_q$ denotes the ℓ^q-norm, $q > 0$ is an exponent typically living in the interval $[0, 2]$, $\lambda \geq 0$ is a data fitting parameter and $G(\mathbf{d})$ is a regularization term promoting specific properties for the displacement $\mathbf{d}$ such as boundedness and smoothness. The function F measures the discrepancy between the odd lines and the lines directly above. We could also add the term $\lambda\|u(\cdot, j+d_j) - \mathbf{v}_{j-1}\|_q^q$ in the definition of F_j to control the distance between the top and bottom neighbors. This data fitting term was first suggested in [7]. An empirical study led to the conclusion that values of q in the interval $[0.5, 1]$ provided the best reconstruction results.

The model (3) is a natural approach to estimate the displacement $\mathbf{d}$. It however suffers from two flaws. First the mapping $t \mapsto u(\cdot, j+t)$ is nonlinear, making the energy (3) nonconvex, even for $p \geq 1$. Hence, nonlinear programming methods can result in spurious local minimizers. Second, we are interested in a setting where n is very large, making the resolution numerically involved. The main contributions of this work is to propose more efficient numerical alternatives.

3.2. A dynamic programming solver

Consider a classical H^1 regularization:

$$G(\mathbf{d}) = \sum_{j=2}^{n} G_j(d_j, d_{j-1}) \text{ with } G_j(x, x') = \lambda(x-x')^2.$$

The problem (3) can be then be attacked by using dynamic programming, which consists in breaking the initial problem in iteratively solving a sequence of simpler sub-problems. Two possibilities are available: either we discretize the shifts and apply discrete dynamic programming [11], or we don't and we can then turn to more recent continuous dynamic programming [12], which is currently restricted to the cases $p = 2$ or $p = 1$. In this work, we use the former by imposing the displacements to live on a grid of step-size $\delta = 1/10$. In what follows, we let $s = \lfloor 2R/\delta \rfloor$ denote the number of possible labels for d_j.

For this problem, we can sequentially construct the arrays:

$$\begin{aligned} &E_1(d_1) = F_1(d_1) \\ &E_j(d_j) = F_j(d_j) + \min_{d_{j-1}}(E_{j-1}(d_{j-1}) + G_j(d_j, d_{j-1})) \end{aligned}$$

The energy E_j can be interpreted as the optimal cost function restricted to the first j-terms, conditionally to the fact that the j-th element has value d_j. The global minimizer of (3) can then be computed iteratively as:

$$\begin{aligned} &d_n^* = \operatorname*{argmin}_{d_n} E_n(d_n) \\ &d_j^* = \operatorname*{argmin}_{d_j} E_j(d_j) + G_j(d_j, d_{j+1}^*), j = n-1, \ldots, 1. \end{aligned}$$

This approach has a complexity in $O\left(\max(mns, ns^2)\right)$. The cost is dominated by the construction of the n tables E_j.

Computing $F_j(d_j)$ for the s possible d_j has a complexity $O(ms)$, while computing the minimum for each d_j has a complexity in $O(s^2)$.

3.3. A convexification method

The previous method is intrinsically sequential: the tables E_j are constructed one after the other. Hence the parallelization can only be used to evaluate the elementary functions F_j of cost $O(m)$. The huge progress of massively parallel architectures make it tempting to consider faster alternatives. We propose one based on a convex relaxation below.

Weight optimization The interpolation formula (2) reads

$$u(\cdot, j + d_j) = \psi(k - d_j)\mathbf{u}_{j+k} + \psi(k + 1 - d_j)\mathbf{u}_{j+k+1}$$

for $k = \lfloor d_j \rfloor$. In order to explain the relaxation, first remark that the above equation can be rewritten as

$$u(\cdot, j + d_j) = \sum_{k=-R-1}^{R+1} w_{k,j}\mathbf{u}_{j+k}, \tag{4}$$

with $w_{j,k} = \psi(k - d_j)$ and using the convention $u_{i,k} = 0$ for $k \notin [\![1, n]\!]$. Above, we used the facts that $|d_j| \leq R$ and that the interpolation kernel ψ is compactly supported to limit the range of k to $[\![-R - 1, R + 1]\!]$.

Let $p = 2R + 3$ and $\mathbf{w} = [\mathbf{w}_1, \dots, \mathbf{w}_n] \in \mathbb{R}^{p\times n}$ with $\mathbf{w}_j = (w_{k,j})_{-R-1\leq k\leq R+1}$ denoting the weights associated to the interpolation of the j-th pixel. Let

$$\mathbf{z}_{(j)} = \left[\mathbf{u}_{j-R-1}, \dots, \mathbf{u}_{j+R+1}\right] \in \mathbb{R}^{m/2\times p}, \tag{5}$$

denote a slice of $\mathbf{u}$. A natural approach to look for the interpolation weights $\mathbf{w}$ is to solve the following least squares problems:

$$\inf_{\mathbf{w}_j\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{z}_{(j)}\mathbf{w}_j - \mathbf{v}_j\|_2^2. \tag{6}$$

Notice that the term $\mathbf{z}_{(j)}\mathbf{w}_j$ above corresponds to a matrix-vector product which performs the interpolation. In this formula, we relaxed the constraint that there exists a shift d_j such that $w_{j,k} = \psi(k - d_j)$. This formulation has the following assets: i) the interpolation weights $\mathbf{w}_j$ are optimized independently from each other, ii) they are obtained by solving n linear systems of size $p \times p$, leading to a complexity in $O(np^3)$. It is however unclear whether this formulation allows recovering the shifts. The following proposition answers positively to this question.

Proposition 1. *Assume that $f(2i, x) = f(2i + 1, x)$ for all x and i. Assume that the columns of $\mathbf{z}_{(j)}$ are linearly independent. Then the unique solution of (6) satisfies $\mathbf{w}_j[k] = \psi(k - d_j)$.*

Obviously, the assumption $f(2i, x) = f(2i + 1, x)$ is unrealistic and would make the problem pointless. Proposition 1 can however be refined to encompass differences between adjacent lines, with a robustness to noise that depends on the minimum singular value of $\mathbf{z}_{(j)}^T\mathbf{z}_{(j)}$. We do not detail this point for lack of space. Most importantly, it shows that for $m \geq p$, the formulation (6) should allow to recover the shifts.

In practice, this simple principle works approximately well for $m \gg p$ and allows to correctly estimate a few of the shifts d_j. However, the conditioning of $\mathbf{z}_{(j)}^T\mathbf{z}_{(j)}$ cannot be controlled uniformly over j (the image domain) since it depends on the image contents. This advocates for the use of regularization.

Weights regularization The two natural regularizers for this problem are the following:

- The weights $\mathbf{w}_j$ are nonnegative and sum to 1, i.e.

$$\mathbf{w} \in \mathcal{W} = \{\mathbf{w} \in \mathbb{R}^{p\times n}, w_{k,j} \geq 0, \sum_{k=1}^{p} w_{k,j} = 1, \forall j\}.$$

- The differences $\mathbf{w}_j - \mathbf{w}_{j+1}$ between adjacent weights should be small, suggesting to penalize a term of the form

$$R(\mathbf{w}) = \frac{1}{2}\sum_{j=1}^{n} \|\mathbf{w}_j - \mathbf{w}_{j+1}\|_2^2.$$

These considerations, lead to the following *convex* problem:

$$\inf_{\mathbf{w}\in\mathcal{W}} \sum_{j=1}^{n} \frac{1}{2}\|\mathbf{z}_{(j)}\mathbf{w}_j - \mathbf{v}_j\|_2^2 + \lambda R(\mathbf{w}). \tag{7}$$

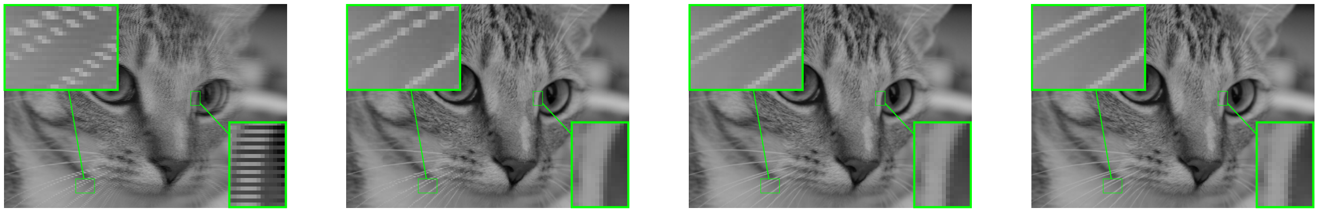
It can be solved globally with standard optimization techniques such as an accelerated projected gradient descent [13]. The downside is that the regularization led to a significant increase of the problem size, since we now work with an $p \times n$ variable instead of n independent problems of size p.

4. RESULTS

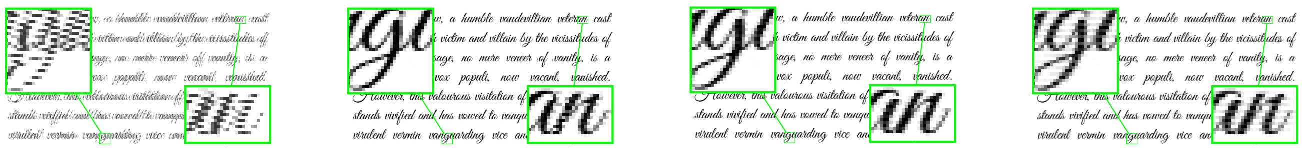
All codes were implemented in Python, running on an Intel Xeon at 2.8Ghz. We used a single core, the default setting under numpy.

To validate the proposed approaches, we simulated 3 different displacement functions $\mathbf{d}$ of amplitude 10.4 pixels (Jit1: constant, Jit2: a parabola, Jit3: a piecewise quadratic displacement) and applied them to two images: a cat of size 440×300 and a text of size 756×756.

Setting $q = 1$ for the DP algorithm, we used the two proposed approaches and the method from [10] to dejitter the simulated shifted images. The results can be compared qualitatively in Fig 2, and quantitatively in Tables 1 and 2 where we report the reconstruction SNR, the computing times, and the SSIM for completeness.

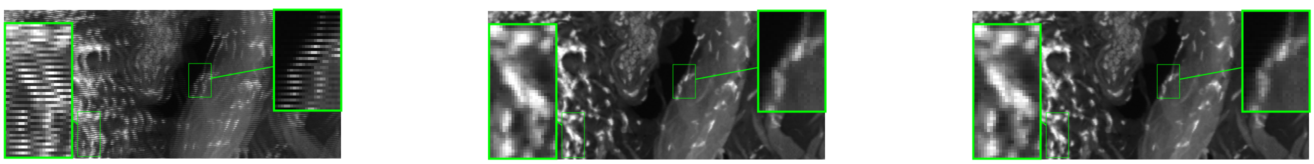


(a) Cat + Jit2 (b) [10] 36.7dB (c) DP 43.2dB (d) Relax 38.2dB



(e) Text+Jit3 (f) [10] 25.0dB (g) DP 30.6dB (h) Relax 26.9dB

Fig. 2. Comparisons of the different algorithms on synthetic data.



(a) Original (b) DP (c) Relax

Fig. 3. Experiment on a crop of a real 10000×15000 image. This is a pancreatic tissue taken with the InnoQuant scanner. The computing times were 226” for DP and 218” for Relax.

We can see that both proposed approaches are at least twice faster than the reference [10]. The dynamic programming approach provides the best signal to noise ratios consistently, usually by a large margin. We conjecture that the times for Relax could be further reduced by one or two orders of magnitude with a clever implementation on a GPU for instance. We did not follow this route since the computing times were already sufficiently small for our application on InnoQuant.

	Jit 1	Jit 2	Jit 3	”	Jit 1	Jit 2	Jit 3	”
[10]	39.9	37.0	33.9	0.46	18.9	22.3	20.3	1.2
DP	**40.7**	**42.7**	**43.2**	**0.11**	**29.5**	**30.6**	**30.7**	**0.36**
Relax	36.8	37.1	38.2	0.25	26.5	26.9	26.1	0.47

Table 1. Reconstruction SNR in dB and mean times in seconds for the cat (left) and the text (right).

	Jit 1	Jit 2	Jit 3	Jit 1	Jit 2	Jit 3
[10]	**.990**	.968	.942	.907	.966	.946
DP	.987	**.991**	**.992**	**.988**	**.991**	**.991**
Relax	.969	.970	.976	.971	.975	.969

Table 2. Reconstruction SSIM for the cat (left) and the text (right).

The results of both methods on real data can be compared qualitatively Fig. 3. Both approaches provide real-time satisfactory results.

5. CONCLUSION

As a conclusion, we proposed two effective and certified approaches to dejitter line scanning microscopy images. The first one is based on dynamic programming and runs in linear time. On our test examples, it seems to provide the best results in terms of image quality. The second one is based on a relaxation approach, leading to a convex problem. The numerical complexity is harder to describe, but it seems to provide a better alternative if time is the first concern. Both approaches are robust since they provide globally optimal solutions, contrarily to more standard nonlinear programming approaches. We believe that a parallelization on the GPU could lead to a real-time solution even for very large images. We leave this aspect for a future prospect. The codes are freely available at `https://github.com/pierre-weiss/line_scanning_microscopy_innopsys`.

Acknowledgments P. Weiss was supported by the ANR Micro-Blind and from ANR-3IA Artificial and Natural Intelligence Toulouse Institute.

6. REFERENCES

[1] Daniel Bec, Vincent Paveau, and Stephane Le Brun, "Fluorescence-based scanning imaging device," Mar. 22 2011, US Patent 7,911,670.

[2] Dayong Jin, Yiqing Lu, and James Austin Piper, "Two-directional scanning for luminescence microscopy," May 28 2015, US Patent App. 14/401,103.

[3] Georg Schuele and I Phillip Gooding, "Laser eye surgery system," Dec. 26 2017, US Patent 9,849,032.

[4] Anil C Kokaram, "Line registration for jittered video," in *Motion Picture Restoration*, pp. 99–118. Springer, 1998.

[5] Anil C Kokaram, *Motion picture restoration: digital algorithms for artefact suppression in degraded motion picture film and video*, Springer Science & Business Media, 2013.

[6] Louis Laborelli, "Removal of video line jitter using a dynamic programming approach," in *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*. IEEE, 2003, vol. 2, pp. II–331.

[7] Mila Nikolova, "One-iteration dejittering of digital video images," *Journal of Visual Communication and Image Representation*, vol. 20, no. 4, pp. 254–274, 2009.

[8] Guozhi Dong, Aniello Raffaele Patrone, Otmar Scherzer, and Ozan Öktem, "Infinite dimensional optimization models and pdes for dejittering," in *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, 2015, pp. 678–689.

[9] Lukas F Lang, "A dynamic programming solution to bounded dejittering problems," in *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, 2017, pp. 146–158.

[10] Hoai-Nam Nguyen, Vincent Paveau, Cyril Cauchois, and Charles Kervrann, "A variational method for dejittering large fluorescence line scanner images," *IEEE Transactions on Computational Imaging*, vol. 4, no. 2, pp. 241–256, 2018.

[11] Pedro F Felzenszwalb and Ramin Zabih, "Dynamic programming and graph algorithms in computer vision," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 4, pp. 721–740, 2010.

[12] Vladimir Kolmogorov, Thomas Pock, and Michal Rolinek, "Total variation on a tree," *SIAM Journal on Imaging Sciences*, vol. 9, no. 2, pp. 605–636, 2016.

[13] Patrick L Combettes and Jean-Christophe Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*, pp. 185–212. Springer, 2011.