



**HAL**  
open science

# A variational method for dejittering large fluorescence line scanner images

Hoai-Nam Nguyen, Vincent Paveau, Cyril Cauchois, Charles Kervrann

► **To cite this version:**

Hoai-Nam Nguyen, Vincent Paveau, Cyril Cauchois, Charles Kervrann. A variational method for dejittering large fluorescence line scanner images. *IEEE Transactions on Computational Imaging*, 2018, 4 (2), pp.16. hal-01577139

**HAL Id: hal-01577139**

**<https://inria.hal.science/hal-01577139>**

Submitted on 24 Aug 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# A Variational Method for Dejittering Large Fluorescence Line Scanner Images<sup>1</sup>

Hoai-Nam Nguyen<sup>1</sup>, Vincent Paveau<sup>2</sup>, Cyril Cauchois<sup>2</sup>, Charles Kervrann<sup>1</sup>

<sup>1</sup> Inria, Centre Rennes-Bretagne Atlantique, SERPICO Project Team, 35042 Rennes, France

<sup>2</sup> Innopsys Inc., Carbonne 31390, France

---

## Abstract

We propose a variational method dedicated to jitter correction of large fluorescence scanner images. Our method consists in estimating a dense displacement field representing the jitter. An energy functional composed of a re-alignment criterion and a differential-based regularizer is especially designed. We adopt the half-quadratic splitting approach to decouple convex and non-convex terms in such a way that the original minimization problem can be solved by alternating two optimization steps – one using a proximal algorithm for the convex part and the other using an exhaustive search for the non-convex part. Experiment results on artificial and large real fluorescence images demonstrate that our method is not only capable to handle large displacements but is also efficient in terms of sub-pixel precision without modifying image intensities when comparing with image regularization-based approaches.

**Keywords :** Dejittering, fluorescence scanner, variational method, regularization, optimization, quadratic relaxation, proximal algorithm

## 1 Introduction

Over the few past decades, the evolution of digital technology has radically changed the way in which images are handled, stored, visualized and transmitted. Nowadays, most of imaging devices are able to transform acquired images into digital form in order to allow not only the recording on electronic medium, but also the processing by computers. This transformation is called digitization which consists in generating a finite sequence of samples from the input (analog) signal. Despite number of advantages, digitization often suffers from undesirable artifacts during the acquisition, transmission and recording processes due to imperfections of one or more components of the involved devices. One of those artifacts is jitter – a type of distortion characterized by the mispositioning of pixels from their true position.

There are several situations where jittering frequently arises. A typical example is the analog-to-digital conversion of video from old magnetic tapes in which horizontal displacement of image lines (line jitter) might happen ; and as a result, edges appear to be jagged in vertical direction as illustrated in the first row of Fig. 1. It is mainly because the synchronization signals that contain

---

1. This work was supported by Innopsys Inc.

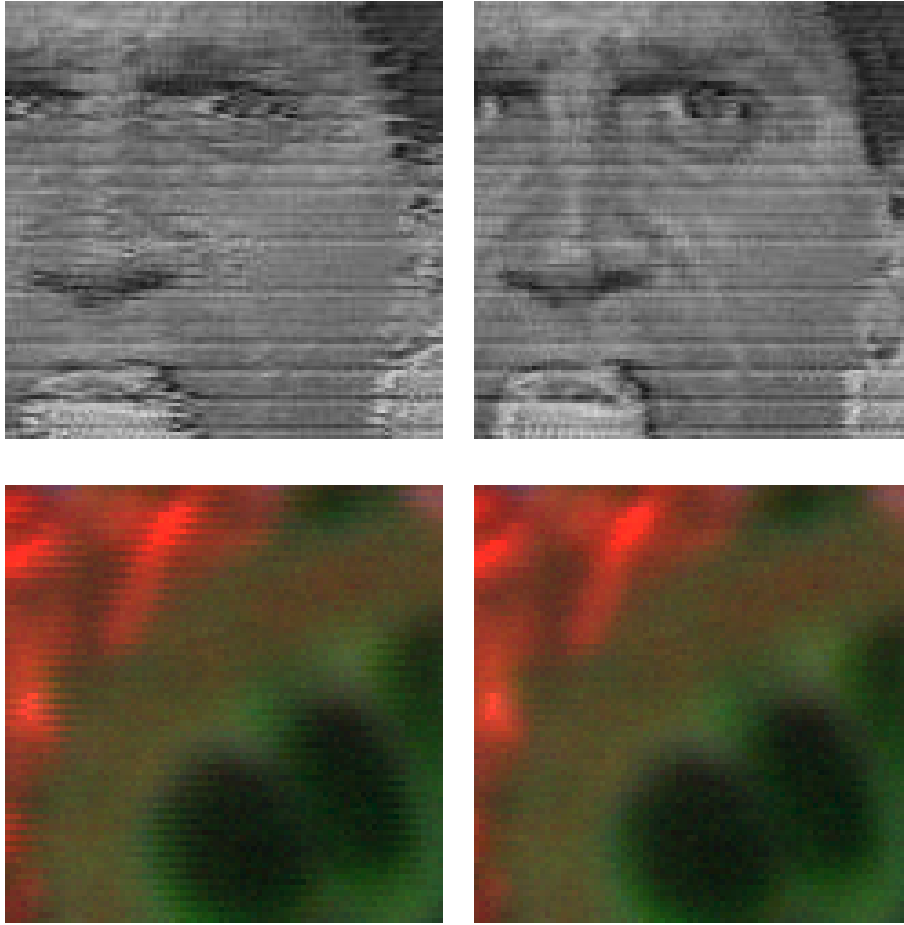


FIGURE 1 – **Examples of jitters.** From top to bottom : jitter arising in analog-to-digital video conversion (Source : <http://www.mee.tcd.ie/~ack/cd/linereg/linereg.htm>) and in fluorescence line scanner (by the courtesy of Innopsys Inc.). From left to right : jittered and de-jittered image.

the information of the relative position of image rows to each other were corrupted by the noise in the video signal and/or by the degradation of the storage medium. Line jitter can also occur during wireless video transmission due to delayed and loss packets. Another case is the video interlacing technique that scan video images by recording two fields at half vertical resolution at two different times and alternating lines of these fields for doubling the frame rate (temporal resolution) of a video while remaining the same bandwidth (amount of data). The temporal difference between recorded fields can lead to a rolling effect in both horizontal and vertical directions as a consequence of relative motions of the camera and the objects in the scene. Similarly, during long acquisition process, random displacement (vibration) of the imaging device and the imaged object is a major cause of jittering in biomedical equipment such as CT (computer tomography) and MRI (magnetic resonance imaging) scanners.

In this paper, we do not address the problem of jittering in general, but we focus only on a particular case of jitter arising in dedicated fluorescence line scanners<sup>2</sup>, as illustrated in the second row of Fig. 1. Here, the images are acquired pixel by pixel along each line with a change of scan

---

2. InnoScan 1100 AL developed by Innopsys Inc., for more details of the product please refer to <https://www.innopsys.com/en/lifesciences-products/microarrays/innoscan/innoscan-1100-al>

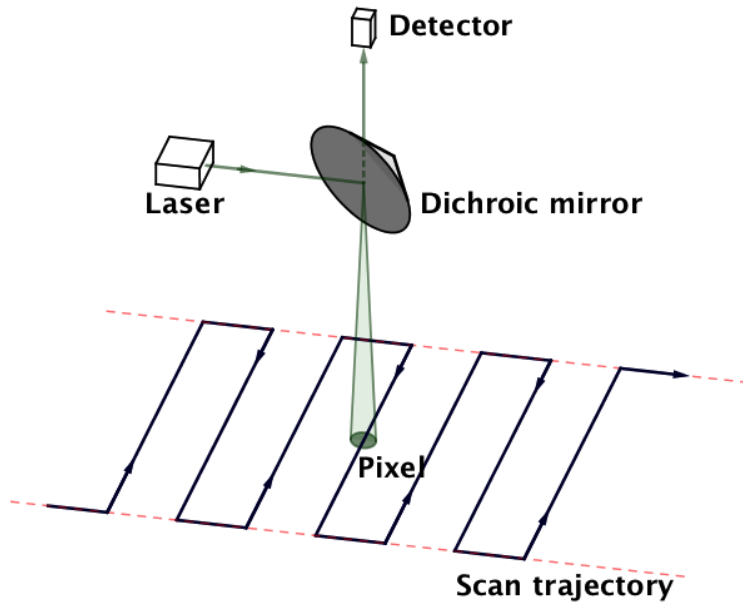


FIGURE 2 – **Image acquisition process involved in fluorescence line scanners.** Images are acquired pixel by pixel along each line with a change of scan direction between two subsequent lines.

direction between two subsequent lines (see Fig. 2). Such an acquisition system requires a very accurate positioning of each acquired pixel in order to provide output images without distortion. However, because of i/ imperfect synchronization between mechanical and electronic components, ii/ small objective rotations induced by abrupt acceleration and deceleration during scanning and iii/ spatially-varying sinusoidal scan speed, the pixels on subsequent lines are horizontally displaced in opposite directions, resulting in a structured jitter (see Figs. 1, 3 and 10). Up to our knowledge, there is unfortunately no existing method to compensate the displacement errors on such jittered fluorescence images. Accordingly, we propose a computational approach inspired from variational optical flow methods. Our approach consist in minimizing an energy functional which combines explicitly a non-convex data term with a convex spatial regularity term with respect to the horizontal displacement while taking into account the special structure of the jitter. This optimization problem can be solved efficiently by introducing an auxiliary variable and a quadratic constraint to decouple the data and the regularity term into two sub-problems in the way that one of which is convex and the other one is non-convex but does not exhibit spatial dependency. The proposed decoupling allows to solve each sub-problem separately with two appropriate algorithms : a proximal algorithm for the convex sub-problem and an exhaustive search for the non-convex sub-problem. The minimization of the proposed energy can be therefore performed by alternating the two optimization steps. Experiments on both simulated and real images show that our approach can handle large and non-integer displacements, which are not addressed in the majority of existing dejittering methods, while avoiding the over-smoothing effect of image-regularization based approaches.

The remainder of the paper is organized as follows. In the next section, several existing methods for image dejittering and deinterlacing are reviewed. In Section 3, we present the proposed acquisition model based on the sampling theory the relationship between the ideal continuous image and its discrete jitter-free as well as its jittered version. We also define the energy functional and propose an algorithm to remove jitter artifacts. Technical details of the proposed algorithm are

given in Section 4. Section 5 presents the experimental results obtained on simulated and real data. Finally, the last section gathers the conclusions drawn from this work and gives a perspective to further improvements.

## 2 Related works for image dejittering and deinterlacing

Analog video signals have a specific format that defines how the timing reference (also known as “time base”) is embedded. It allows the receiving device to correctly extract the necessary synchronization information for reconstructing images and providing visual display. The corruption or even the loss of this timing reference consequently results in an extraction of incorrect synchronization information and thus introduces jitter. In the late of 80’s, “time base correction” techniques were already used for restoring the embedded synchronization information in the analog signal by trying to remove noise on the non-picture part of the signal in order to reproduce an accurate display of the input video. However, such a technique which is able to process solely analog signals cannot be applied to current digital videos. Over the last two decades, intrinsic dejittering [1] techniques were introduced as an alternative solution. Indeed, these techniques which rely only on jittered image data are much more flexible and widely applicable when compared with conventional (non intrinsic) “time base correctors”.

To perform (intrinsically) image dejittering, one can envisage simple approaches such as line averaging and correlation matching between two subsequent lines. Unfortunately, the former approach tends to produce blur and decreases the vertical resolution, whereas the latter tends to cause a bias in vertical direction due to its local property. At the end of 90’s, Kokaram et al. invented the first intrinsic method [1, 2], in which the authors assumed jitter-free images obey a 2D-AR (autoregressive) model. The unknown AR coefficients and the line displacements are jointly estimated by blocs using an iterative algorithm. Whitening of the estimated displacement can be performed in the case of drift (i.e. restored images are warped) to remove the low frequency component of the estimation.

Later, Laborelli [3] proposed a different approach to estimate the line displacement by considering the  $L_1$  norm of the difference of two or three consecutive shifted lines as a local criterion and using it to define a global cost which is optimized by dynamic programming. Right afterwards, Shen described in [4] a joint denoising and dejittering algorithm which consists in minimizing a energy functional in the Bayesian framework with respect to the original (unjittered) image and the associated displacement. The considered energy is derived from the posterior probability of the unjittered image and the displacement given the observed jittered image. This conditional probability is defined under the Gaussian distribution assumption of the displacement and of the noise, combined with the BV (bounded variation) image model. Instead of considering such a joint denoising and dejittering approach, Kang and Shen developed a two-step displacement-estimation method called “Bake and Shake” [5] which relies on Perona-Malik diffusion [6] to reduce the noise and the rolling effect of jittered images. The key idea of this method is that the intermediate diffused (baked) image – less noisy and less jittered – can guide the estimation of the displacement (i.e. the “shake” step). Later on, the “Bake and Shake” authors introduced in [7] the notion of slicing moments of BV images – a measure to quantify how much the image is jittered. Using this notion, the displacement of image lines is obtained by minimizing a  $L_2$ -TV (total variation)-based functional of the vertical slicing moments.

In contrast to these continuous-optimization-based methods, Nikolova proposed a fast algorithm for image dejittering [8, 9] by performing a complete search over a finite set of allowed integer values of shifts to optimize a non smooth and (eventually) non-convex local criterion. This local criterion is

an extended version of the criterion mentioned in [3] by using normalized  $L_p$  norms (with  $p = 0.5$  or  $p = 1$ ) and possibly higher order criteria (e.g. more than three adjacent lines). Recently, Lenzen et al. introduced a PDE (partial differentiable equation) derived via semi-groups from a nonconvex energy functional for restoring directly jittered images without explicit estimation of the displacement [10]. This model which is related to the mean curvature flow [11, 12] corresponds to the minimization of an energy composed of a linearized re-alignment criterion and a TV regularization. Based on these previous works, Dong et al. proposed an unified class of methods for different types of jitter [13] by generalizing the Nikolova’s algorithm to infinite dimensional framework and establishing its relation with the Lenzen’s PDEs.

In the comparison with “line dejittering” which corrects the shift between consecutive lines of a jittered image, deinterlacing is the process of reconstructing a complete image frame from two recorded fields at half vertical resolution at two different times. It consists generally in compensating the motion between these recorded fields due to temporal difference. Inspired from optical flow models, the estimation of the underlying motion have been recently used in state-of-the-art deinterlacing methods. Most of these motion-estimation-based methods are defined in a variational framework with TV regularization [14, 15, 16].

Irrespective of deinterlacing or dejittering algorithms, we can classify the majority of them into three categories : (a) simultaneous approach as [4, 16] which consists in estimating jointly the displacement and the jitter-free original image; (b) displacement estimation methods which represent the majority part on the literature as in [2, 1, 3, 5, 7, 8, 9, 14]; (c) image correction without estimating the displacement as in [15, 10, 13] which is mainly based on the BV image model and variational methods. However, there are problems which are not fully addressed such as large and non-integer displacement, over-fitting, or over-smoothing.

### 3 Jitter modeling : Application to fluorescence line scanner images

In this section, we present a general image acquisition model which explains the relationship between the ideal (undistorted) continuous image and its digitized versions. The digitized images are nothing else than finite sets of samples of the continuous image according to a sampling grid. In this work, we focus on gray-scale bi-dimensional (2D) images for the sake of clarity.

#### 3.1 Notations and image sampling

Let  $u : \Omega \rightarrow \mathbb{R}$  be an undistorted continuous image which is defined on a rectangular domain  $\Omega = [0, M] \times [0, N] \subset \mathbb{R}^2$ . Digitization consists in sampling the input continuous images with respect to a lattice (sampling grid) in order to provide an output discrete image. In the case of an ideal sampling, the sampling grid is a regular grid which is characterized by a couple of orthogonal unit vectors and a vertex-to-vertex distance (also sampling step). Without loss of generality, one can assume that this regular sampling grid is the Cartesian grid  $\mathbb{Z}^2$  (*i.e.* the couple of characterized vectors is the canonical basis  $(\mathbf{e}_1, \mathbf{e}_2)$  of the plane  $\mathbb{R}^2$  and the sampling step equals to 1 (implying that each vertex of the sampling grid is integer point). The sampled version of  $u$  according to  $\mathbb{Z}^2$  which is defined as the set of finite samples  $\{u_{i,j} \triangleq u[i, j]\}_{(i,j) \in \mathbb{Z}^2 \cap \Omega}$  is called the (discrete) jitter-free image. To shorten the notations, we denote the image grid  $\Lambda = \mathbb{Z}^2 \cap \Omega = \{0, 1, \dots, M\} \times \{0, 1, \dots, N\}$ .

Let us denote  $\mathbf{w} : (i, j) \in \Lambda \mapsto (w_1(i, j), w_2(i, j)) \in \mathbb{R}^2$  the dense vector field which represents the jitter occurring during the image acquisition process. The jittered version of  $u$  with respect to

the displacement field  $\mathbf{w}$  is therefore defined as :

$$\left\{ f_{i,j} \triangleq f[i,j] = u[i - w_1(i,j), j - w_2(i,j)] \right\}_{(i,j) \in \Lambda} . \quad (1)$$

In other words,  $\{f_{i,j}\}_{(i,j) \in \Lambda}$  is a sampled version of  $u$  according to a irregular sampling grid

$$\tilde{\Lambda} = \{(i - w_1(i,j), j - w_2(i,j)), \forall (i,j) \in \Lambda\} \subset \mathbb{R}^2 . \quad (2)$$

The restoration of  $\{u_{i,j}\}_{(i,j) \in \Lambda}$  from irregular samples  $\{f_{i,j}\}_{(i,j) \in \Lambda}$  with unknown sampling position  $\tilde{\Lambda}$  is the so-called *dejittering* procedure.

### 3.2 Alternating line pixel jitter

In contrast to a general irregular sampling case in which the perturbed sampling position  $\tilde{\Lambda}$  does not need to have any particular structure (beyond a minimal sampling density) [17, 18], dejittering usually exploits the structure of the associated vector field  $\mathbf{w}$  to compensate the displacement errors and re-align the jagged edges. In this work, we focus on a specific jitter arising in images acquired by fluorescence line scanners called (InnoScan 1100 AL). A typical example of jittering is illustrated in Fig. 3 showing a checkerboard pattern in which the displacement between two subsequent image lines is not constant along the horizontal axis. This situation is quite different from the usual case of line jitter which is well documented in the literature [1]-[13].

On such jittered images, we remark that : (i) the pixels on the even lines are shifted in the opposite direction with respect to those on the odd lines ; and (ii) there is no apparent displacement between lines of the same set of even lines or odd lines. More specifically, the rolling effect appears only in the vertical direction and not in the horizontal direction. By measuring the gap between vertical contours on subsequent lines at difference positions in the jittered images, we also notice that the magnitude of displacement along horizontal axis varies smoothly according to a specific pattern depending on pixel positions (see Fig. 3). The observed jitter can be named as ‘‘alternating line pixel jitter’’. Based on the described observations, we propose to model this jitter by a vector field  $\mathbf{w} = \{\mathbf{w}(i,j) = (w_{i,j}, 0)\}_{(i,j) \in \Lambda}$  whose the vertical component is zero and the horizontal component satisfies the two following conditions :

$$\begin{cases} w_{i,j} = 0 \text{ if } j \in 2\mathbb{Z} , \\ w_{i,j} w_{i',j} \geq 0 \text{ for any } (i, i') \in \mathbb{Z}^2 . \end{cases} \quad (3)$$

The first condition implies the alternating structure of the observed jitter (one line of every two line is shifted). The second condition implies that the horizontal component of the displacement vector at a point on the a given line has the same sign as those at another point on the same line – resulting in the same displacement direction on the same line. Since the vertical component of  $\mathbf{w}$  is assumed to be constant, we estimate only the horizontal component  $w : (i,j) \mapsto w_{i,j}$  to perform image dejittering.

## 4 Variational method for displacement estimation

### 4.1 Energy functional

One strategy of dejittering is to estimate the underlying displacement in order to remove rolling effect due to the jitter. To deal with the ‘‘alternating line pixel jitter’’ occurring in fluorescence scanner images, we aim at estimating the scalar field  $w$  which represent the displacement in horizontal

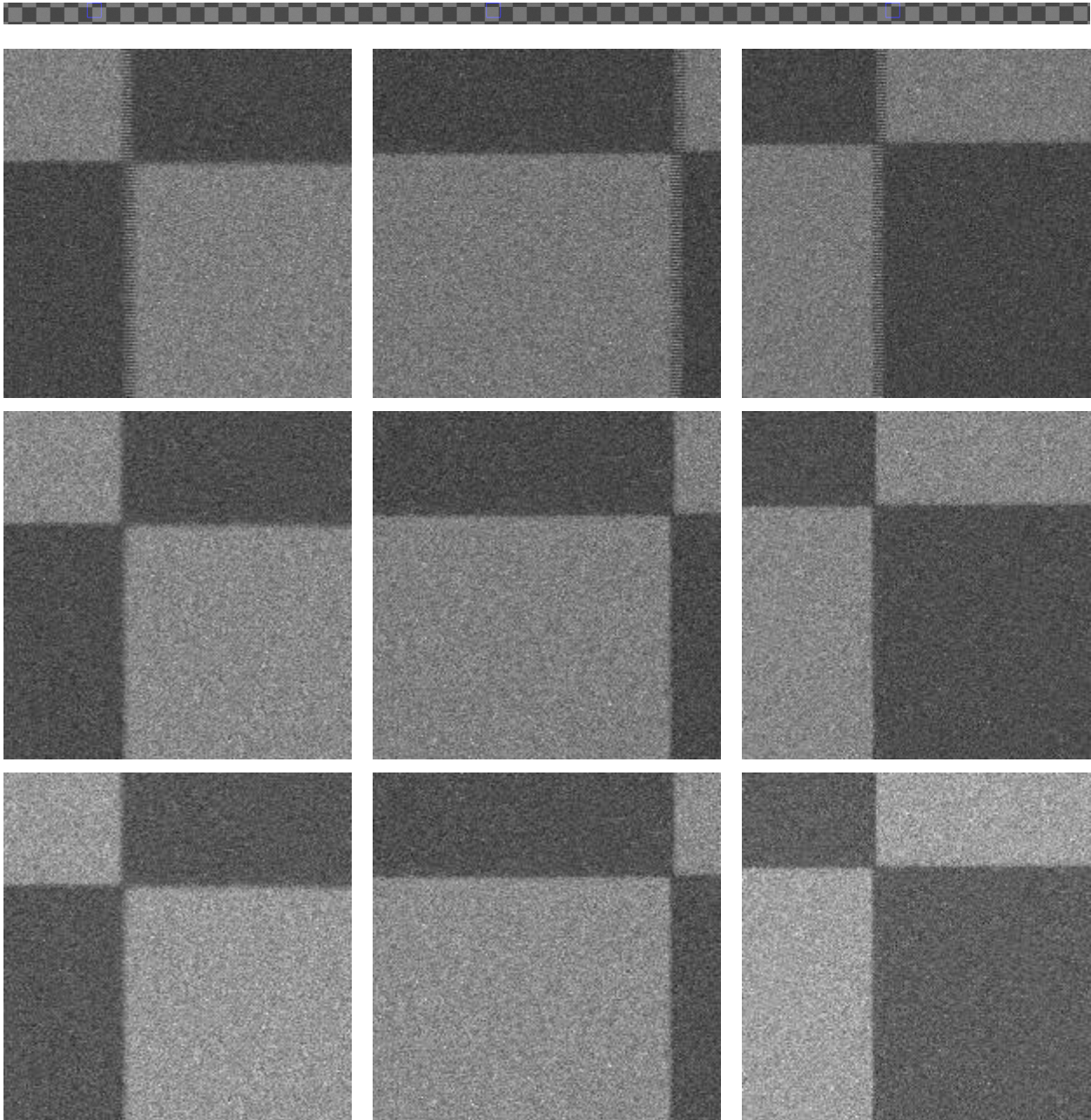


FIGURE 3 – Scanned image of a tool slide used for calibrating scanners before and after the correction of vertical rolling effect using our algorithm. First row : very large image with a width of 22780 pixels (22.78 millimeters in real scale). Second row : zoom on selected areas (blue boxes in the full-width image) with different values of displacement. In the middle, the maximal estimated displacement is 8 pixels ; whereas, in the remaining cases, this value is about 6(4) pixels. Third row : corrected by our method without regularization ( $\lambda = 0$ ). Fourth row : corrected with regularization.



direction by minimizing a global energy of the following form :

$$E(w) = \underbrace{\sum_{i=0}^M \sum_{j=0}^N \rho(i, j, f, w)}_{E_D(w, f)} + \underbrace{\sum_{i=0}^M \sum_{j=0}^N \phi(Lw_{i, j})}_{E_R(w)} , \quad (4)$$

where  $L$  is a linear operator used to control the spatial regularity of  $w$ . The energy (4) combines explicitly a potential  $\rho(\cdot)$ , which penalizes irregular shapes of the image  $f$  on the neighborhood of a pixel  $(i, j)$ , with a regularization potential  $\phi(\cdot)$  which penalizes high values of the norm of  $Lw_{i, j}$ . This reformulation is inspired from the concept of regularized vector field which is widely used in optical flow to compute the motion between different frames in a video sequence [19, 20]. In context of dejittering, the displacement regularization allows to overcome not only the problem of data over-fitting of unregularized approaches [1, 9] but also the effect of over-smoothing of approaches based on image regularization as described in [4, 10, 13].

## 4.2 Data fidelity term

By definition, the corrected version of the jittered image  $f$  by the displacement  $w$  is written as :

$$\{\tilde{u}_{i, j} = f[i + w_{i, j}, j]\}_{(i, j) \in \Lambda} . \quad (5)$$

By considering the vertical regularity of the corrected image  $\tilde{u}$ , a data potential  $\rho(\cdot)$  is defined to re-align the jittered image  $f$  as :

$$\rho(i, j, f, w) = |D^{(k)}\tilde{u}_{i, j}|^\beta , \quad (6)$$

where  $D^{(k)}$  denotes the discrete  $k^{\text{th}}$  order vertical derivative operator and  $\beta$  is a positive parameter which controls the amount of non zero entries of  $D^{(k)}\tilde{u}$ . In other words,  $\rho(\cdot)$  favors sparse  $D^{(k)}\tilde{u}$  if  $0 < \beta \leq 1$ , otherwise this potential promotes non-sparse configurations. If  $k = 1, 2$  and  $3$ , we consider the following discrete derivatives :

$$\begin{aligned} D^{(1)}\tilde{u}_{i, j} &= \tilde{u}_{i, j} - \tilde{u}_{i, j-1} \\ &= f[i + w_{i, j}, j] - f[i + w_{i, j-1}, j - 1] , \\ \\ D^{(2)}\tilde{u}_{i, j} &= \tilde{u}_{i, j+1} - 2\tilde{u}_{i, j} + \tilde{u}_{i, j-1} \\ &= f[i + w_{i, j+1}, j + 1] - 2f[i + w_{i, j}, j] \\ &\quad + f[i + w_{i, j-1}, j - 1] , \\ \\ D^{(3)}\tilde{u}_{i, j} &= \tilde{u}_{i, j+1} - 3\tilde{u}_{i, j} + 3\tilde{u}_{i, j-1} - \tilde{u}_{i, j-2} \\ &= f[i + w_{i, j+1}, j + 1] - 3f[i + w_{i, j}, j] \\ &\quad + 3f[i + w_{i, j-1}, j - 1] - f[i + w_{i, j-2}, j - 2] , \end{aligned} \quad (7)$$

with the symmetric boundary conditions in the vertical direction such that  $\tilde{u}_{i, j} = \tilde{u}_{i, -j}$  if  $j < 0$  and  $\tilde{u}_{i, j} = \tilde{u}_{i, 2N-j}$  if  $j > N$ . In what follows, we choose  $k = 2$  and  $\beta = 1$  since these values allow to recover images with nearly piece-wise linearity in the vertical direction (see [8, 9]). However, this potential exhibits spatial dependency on  $w$  since  $\rho(i, j, f, w)$  not only depends on  $w_{i, j}$  but also on other entries  $w_{i, j'}$  where  $j'$  is a neighbor of  $j$  according to the order  $k$  of the derivative.

Instead of considering the potential (6), (7) which is difficult to optimize globally, a point-wise term is desired in order to avoid spatial dependency on  $w$  and to facilitate the optimization

task. From the condition (3), the displacement field  $w$  is non zero on odd lines (*i.e.* the function  $i \mapsto w_{i,j} \neq 0$  for  $j \notin 2\mathbb{Z}$ ). Therefore, let us consider the odd line translation  $\mathcal{T}_{w_{i,j}}$  defined as :

$$(\mathcal{T}_{w_{i,j}}f)_{k,l} = \begin{cases} f[k, l] & \text{if } l \in 2\mathbb{Z} , \\ f[k + w_{i,j}, l] & \text{otherwise .} \end{cases}$$

By assuming that the variation of the displacement between odd and even lines on a small neighborhood of a point is infinitesimal, we propose a novel non-local potential  $\rho_\sigma$  defined as following :

$$\rho_\sigma(w_{i,j}, f) = \begin{cases} 0 & \text{if } j \in 2\mathbb{Z} , \\ \sum_{(k,l) \in \sigma(i,j)} |D^{(k)}(\mathcal{T}_{w_{i,j}}f)_{k,l}|^\beta & \text{otherwise ,} \end{cases}$$

where  $\sigma(i, j) = \{i - \lfloor \frac{r}{2} \rfloor, \dots, i + \lfloor \frac{r}{2} \rfloor\} \times \{j - \lfloor \frac{s}{2} \rfloor, \dots, i + \lfloor \frac{s}{2} \rfloor\}$  is a patch of size  $r \times s$  pixels centered at  $(i, j)$ . The data fidelity term associated with the potential  $\rho_\sigma(\cdot)$  is :

$$E_D(w, f) = \sum_{i=0}^M \sum_{j=0}^N \rho_\sigma(w_{i,j}, f) . \quad (8)$$

### 4.3 Regularization term

To exploit the alternating structure of the displacement field and impose its smoothness with no sharp discontinuity, we consider the following regularization term :

$$E_R(w) = \sum_{i=1}^M \sum_{j=2}^N \left\{ \lambda (w_{i,j} - w_{i-1,j})^2 + \nu (w_{i,j} - w_{i,j-2})^2 \right\} , \quad (9)$$

where  $\lambda, \nu \geq 0$  are regularity parameters which control the regularity of  $w$  along the horizontal and vertical directions respectively. If we denote  $L_{\lambda,\mu}$  the discrete differential-based operator defined as :

$$(L_{\lambda,\mu}w)_{i,j} = \begin{bmatrix} \sqrt{\lambda}(w_{i,j} - w_{i-1,j}) \\ \sqrt{\mu}(w_{i,j} - w_{i,j-2}) \end{bmatrix} \in \mathbb{R}^2 ,$$

for  $1 \leq i \leq M$  and  $2 \leq j \leq N$ , otherwise  $(L_{\lambda,\mu}w)_{i,j}$  is a null vector, the regularization term (9) can be rewritten as :

$$E_R(w) = \sum_{i=0}^M \sum_{j=0}^N \|(L_{\lambda,\mu}w)_{i,j}\|_2^2 , \quad (10)$$

where  $\|\cdot\|_2$  denotes the Euclidean norm.

### 4.4 Optimization of the energy functional

Considering the proposed data fidelity and regularization terms, the energy functional has the following form :

$$E(w) = \sum_{i=0}^M \sum_{j=0}^N \rho_\sigma(w_{i,j}, f) + \|(L_{\lambda,\mu}w)_{i,j}\|_2^2 , \quad (11)$$

in which the first term is convex with respect to  $w_{i,j}$  and the second term is in general non-convex. The combination of convex and non-convex terms in the expression of the energy makes the underlying optimization problem become much more difficult than a classical convex case which can be solved efficiently by number of existing algorithms [21, 22, 23, 24, 25, 26]. This situation is closely similar to the variation formulation of the optical flow problem [27, 20] in which the non-convex data term is usually coupled with a convex regularity term [28, 29]. In addition, both of these terms eventually may be not continuously differentiable. One feasible approach is to linearize the data term, to replace non differentiable functions by their smoothly approximated version and use standard differentiable optimization techniques to minimize the derived energy. To avoid these approximations, we adopt the approach of Steinbrücker *et al.* [30] and propose to optimize a quadratically relaxed version of (11) of the following form :

$$E_\epsilon(w) = \sum_{i=0}^M \sum_{j=0}^N \rho_\sigma(w_{i,j}, f) + \frac{1}{2\epsilon}(w_{i,j} - v_{i,j})^2 + \|(L_{\lambda,\mu}v)_{i,j}\|_2^2, \quad (12)$$

where  $\epsilon$  is a small constant and  $v : (i, j) \mapsto v_{i,j}$  is an auxiliary variable used to approximate  $w$ . The use of auxiliary variable for decoupling convex and non-convex part is also known as the half-quadratic method which was originally introduced by Geman and Yang [31]. This spitting technique is widely used in both linearized and non-linearized optical flow models [28, 29, 30] to decompose the original optimization problem into two smaller sub-problems whose the optimal solution can be computed in an easier way.

Moreover, since the magnitude of displacement does not exceed a certain bound, one can add the convex constraint  $\{\|w\|_\infty = \max_{i,j} |w_{i,j}| \leq w_{\max}\}$  to the relaxed optimization problem (12) with  $w_{\max}$  the maximal absolute value of displacement. Finally, we consider the following optimization problem :

$$(\hat{w}, \hat{v}) = \arg \min_{w,v} \sum_{i=0}^M \sum_{j=0}^N \left\{ \rho_\sigma(w_{i,j}, f) + \mathcal{I}_{\mathcal{C}}(w_{i,j}) + \frac{1}{2\epsilon}(w_{i,j} - v_{i,j})^2 + \|(L_{\lambda,\mu}v)_{i,j}\|_2^2 + \mathcal{I}_{\mathcal{C}}(v_{i,j}) \right\}, \quad (13)$$

where  $\mathcal{I}_{\mathcal{C}}$  denotes the characteristic function of the subset  $\mathcal{C} = [-w_{\max}, w_{\max}] \subset \mathbb{R}$  defined as ( $r \in \mathbb{R}$ ) :

$$\mathcal{I}_{\mathcal{C}}(r) = \begin{cases} 0 & \text{if } r \in \mathcal{C}, \\ +\infty & \text{otherwise.} \end{cases}$$

This problem can be solved by alternating two optimization steps which update either  $w$  or  $v$  at each iteration :

- For  $v$  being fixed, solve

$$\arg \min_w \sum_{i=0}^M \sum_{j=0}^N \left\{ \mathcal{I}_{\mathcal{C}}(w_{i,j}) + \rho_\sigma(w_{i,j}, f) + \frac{1}{2\epsilon}(w_{i,j} - v_{i,j})^2 \right\}. \quad (14)$$

This problem can be solved point-wise, since there are no spatial dependency term for  $w$  in (14). Therefore, the optimal values for  $w_{i,j}$  at every point  $(i, j)$  can be simply computed by an exhaustive search in the discrete set of possible displacements

$$\mathcal{W} = \{-w_{\max}, -w_{\max} + \delta, -w_{\max} + 2\delta, \dots, w_{\max}\},$$

where the step  $\delta$  can be chosen as small as possible to guarantee the sub-pixel precision of the estimated jitter. In practice, for each point  $(i, j)$ , we minimize the function  $w_{i,j} \mapsto \rho_\sigma(w_{i,j}, f) + \frac{1}{2\epsilon}(w_{i,j} - v_{i,j})^2$  over the finite and countable set  $\mathcal{W}$  rather than  $\mathcal{C}$  which is finite but uncountable. This discrete minimization not only helps to avoid the linearization of the data term, which may yield inaccurate results, but also allows to deal with large and non-integer displacements.

– For  $w$  being fixed, solve

$$\arg \min_v \sum_{i=0}^M \sum_{j=0}^N \left\{ \frac{1}{2\epsilon}(w_{i,j} - v_{i,j})^2 + \|(L_{\lambda,\mu} v)_{i,j}\|_2^2 + \iota_{\mathcal{C}}(v_{i,j}) \right\}. \quad (15)$$

Since the constraint set  $\mathcal{C}$  is convex, this problem can be solved by a simple projected gradient descent algorithm summarized as follows :

$$\begin{cases} \tilde{v}_{i,j}^{(k)} &= \frac{1}{\epsilon}(v_{i,j}^{(k)} - w) + (L_{\lambda,\mu}^\top L_{\lambda,\mu} v^{(k)})_{i,j}, \\ v_{i,j}^{(k+1)} &= \text{proj}_{\mathcal{C}}(v_{i,j}^{(k)} - \gamma_k \tilde{v}_{i,j}^{(k)}) \end{cases}, \quad (16)$$

where  $L_{\lambda,\mu}^\top$  denotes the adjoint of  $L_{\lambda,\mu}$  and  $\text{proj}_{\mathcal{C}}$  denotes the projection to the set  $\mathcal{C}$ . To ensure the convergence of projected gradient descent iterations, the descent step  $\gamma_k$  satisfies the usual condition

$$0 < \gamma_k < \frac{2}{4(\lambda + \mu)}. \quad (17)$$

We note that the projected gradient descent algorithm is a particular case of the forward-backward algorithm [32, 24]. This algorithm also belongs to the family of proximal algorithms [26].

## 5 Experiment results

In order to evaluate our variational method, we consider both true and simulated jittered images. The real images are acquired by fluorescence scanners of the series InnoScan commercialized by Innopsys company in which a laser beam is moving according to a programmed trajectory to scan input microscopic slides. In the scanner design, the velocity of the laser beam is not constant during the scanning : it starts slowly at the beginning of a line, then accelerates and reaches the maximum at the middle, and decelerate smoothly until the end where it changes the direction for the next line. The desynchronization between the mechanical system which controls the movement of the scanning laser beam and the electronic system which converts analog signals to digital output leads to the writing of pixels at wrong positions, and thus introduces vertical rolling effect of odd and even lines with non-constant displacement along the horizontal direction, low at the extremities of the lines and maximal at the middle. The occurred jitter is illustrated in Fig. 3 which shows different magnitudes of displacement.

### 5.1 Evaluation on simulated images

To create the same jittering effect as in fluorescence scanner images, we first simulated a dense displacement field which is zero on odd lines and non-constant on even lines such that the displacement with respect to horizontal coordinate is smoothly increasing on the first half of the line

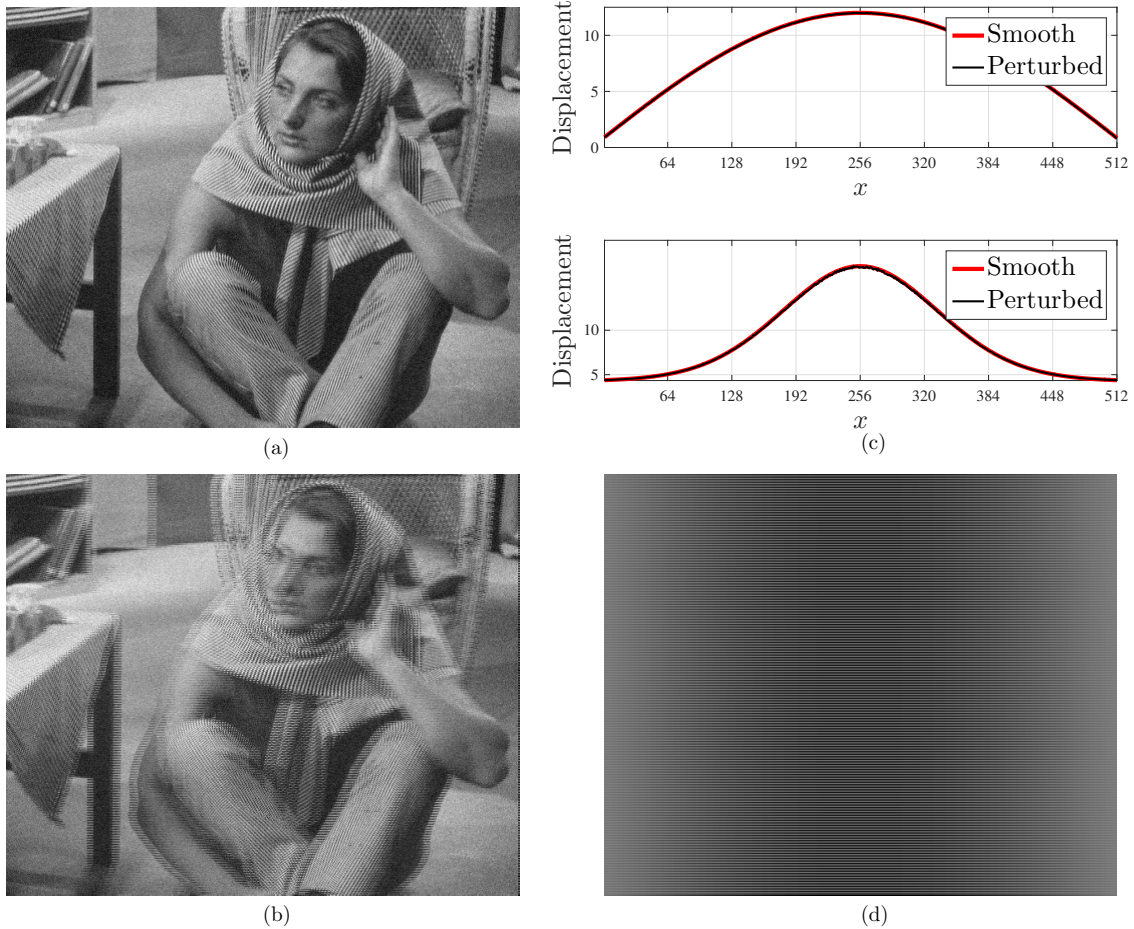


FIGURE 4 – **The simulation of jitter on “Barbara”  $512 \times 512$  image by using sinus function.** (a) Original image. (b) Jittered image. (c) Simulated displacement on even lines with respect to the horizontal coordinate (in pixel) plotted in normal scale (top) and in exponential scale (bottom) for comparison purpose : smooth displacement (red curve) is computed using sinus function and the perturbed version (black curve) is generated by adding small random values. (d) Displacement field associated with (b) (gray is zero value).

and decreasing on the second half by using the sinus function. Random perturbation with small magnitude (less than one tenth of the pixel) was then added to the displacement field to model the micro-vibration of the acquisition system. A set of 8 images was selected as test images which were jittered by using the simulated displacement field. An example of jitter simulation on “Barbara” image is shown in Fig. 4.

### 5.1.1 Robustness to noise

To analyze the robustness to noise of our approach, a white Gaussian noise with zero mean and variance  $\sigma^2$  is added to the original images before being jittered for the purpose of generating noisy and jittered images. The performance of our algorithm on these images is measured by the Peak Signal-to-Noise Ratio (PSNR) and by the Structural Similarity (SSIM) index between the

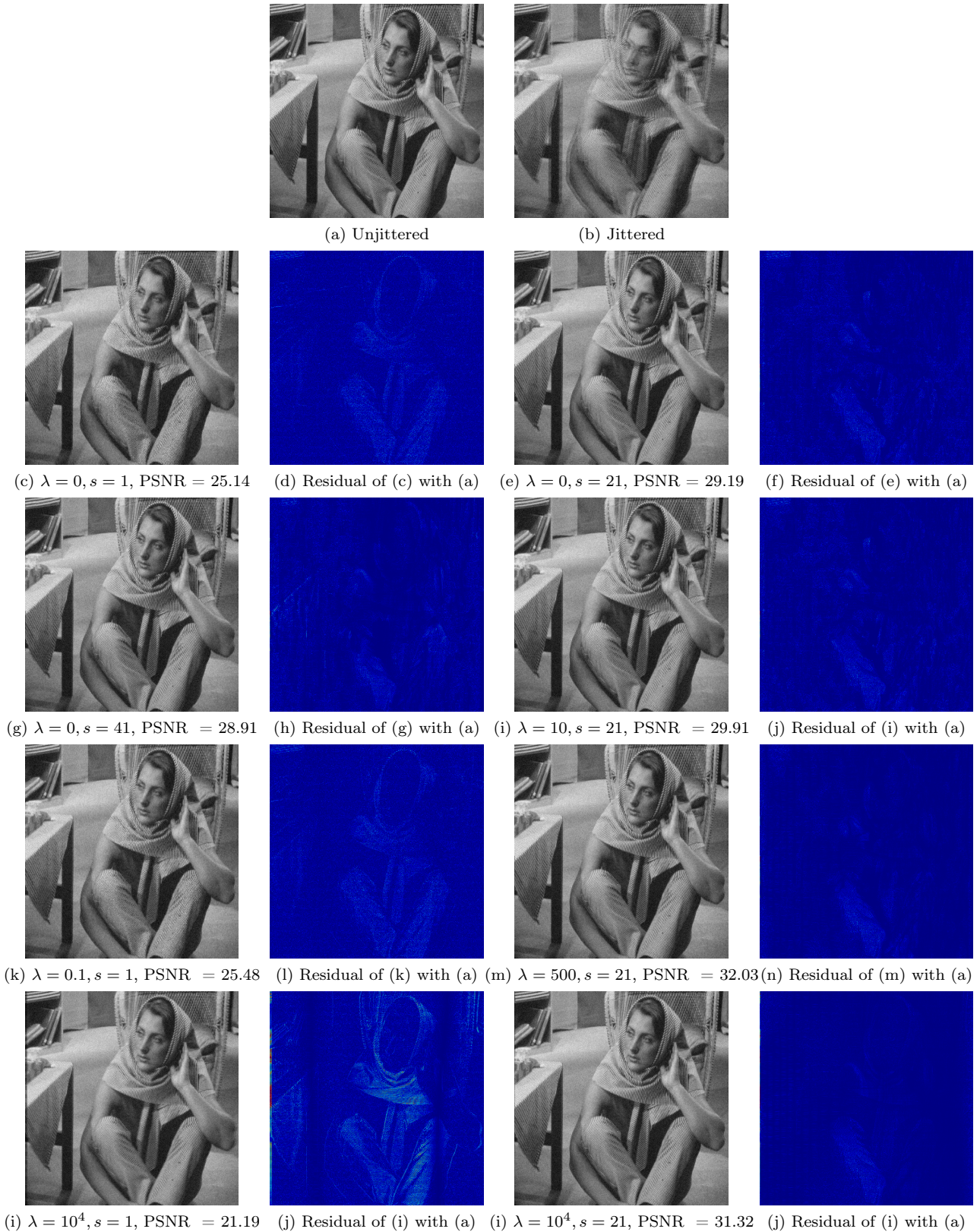


FIGURE 5 – Result of dejittering with different parameters of regularization  $\lambda$  and window size  $s$  on Barbara image ( $512 \times 512$ ) corrupted with white Gaussian noise ( $\sigma = 0.04$ ). Residual images are shown in color where “cold” colors correspond to low residual values and “hot” colors correspond to high residual values.

dejittered images and their unjittered version in both noise-free and noisy cases. For real data in which jitter-free images are unknown, quantitative evaluation as described previously is quite impossible. Therefore, in this case, we only show the dejittering outcome and compare visually in details the obtained results with the input images to highlight the effect of displacement correction. In all our experiments, we normalized images in the range  $[0, 1]$  to avoid the problem of different dynamics of images and produce a homogeneous comparison of the impact of the regularization parameters on dejittering result. For simplification purpose, we used only square windows ( $s = r$ ) and isotropic regularization parameters ( $\lambda = \nu$ ) when performing dejittering on simulated images.

A typical result of our dejittering method with different parameters of regularization and estimation window size is illustrated in Fig. 5. On the second column of Fig. 5, the residual of dejittered images is also displayed to compare the performance of these results. In this example, the jitter-free image was corrupted by a white Gaussian noise with standard deviation  $\sigma = 0.04$  (which corresponds to a value of PSNR = 26.77). The displacement field was estimated by setting the precision of displacement  $\delta = 0.1$  (in pixel) and the parameter  $p = 0.5$ . We can see that even in the case of  $s = 1$  (smallest estimation window) and  $\lambda = 0$  (without regularization) the vertical rolling effect is significantly removed and most of large geometric structures and small details are generally well restored (see Fig. 5(c)-(d) and 6). However, we observe an aliasing effect on restored image with these parameters (see the first column in Fig. 6), this artifact appears generally in regions near the left and right borders of the image where the displacement is relatively lower than those at the middle. Nevertheless, it does not disturb the overall visual display. The aliasing is slightly reduced when increasing the the size of the estimation window  $s$  and/or the regularization parameter  $\lambda$  as depicted in the three bottom rows of Fig. 6. In dense texture regions as shown on the third column of Fig. 6, we remark that the corrected image with parameters  $s = 1$  and  $\lambda = 0$  failed to reconstruct properly very fine details. In fact, by using these parameters, the orientation of the texture was wrongly restored with respect to those on the unjittered image. It is mainly due to the repetition of patterns and the lack of regularity of the estimated displacement field.

We expected that the dejittering with higher values of  $s$  and  $\lambda$  which imply spatial regularities of the displacement field would produce better reconstruction, but we obtained a surprising result. The setting  $s = 21$  (large estimation window) and  $\lambda = 0$  (without regularization) yielded a better restoration than  $s = 1$  and  $\lambda = 0.1$  (highest value of SSIM with  $s = 1$  among  $\lambda \in \{0, 0.1, 1, 10, 500, 10000\}$ , see Table 1) and this restoration is very close to the result obtained with  $s = 21$  and  $\lambda = 10$  as well as the original image (see the third column of Fig. 6). It means that our method provides better results for the larger estimation window than for the higher regularization parameters.

### 5.1.2 Influence of $\lambda$ and $s$ parameters

To better understand the impact of  $\lambda$  and  $s$  in the estimation of the displacement field, the estimated displacement corresponding to the dejittering result in Fig. 5 are shown in Fig. 7. For comparison purpose, we also displayed in Fig. 7(a) the simulated displacement (ground truth) which was used to generate jittered images, and in Fig. 7(b) the estimated displacement with  $\lambda = 500$  and  $s = 21$  which have one of the best value of SSIM scores (SSIM=0.9683) for all possible couple  $(\lambda, s)$ . Finally, the displacement estimated with  $\lambda = 10000$  was illustrated in the last row of Fig. 7 as an extreme case. As expected, the larger estimation window and/or the higher regularization parameter, the more regular displacement field is obtained. Indeed, we can observe that image details visibly appear in the estimated field for  $\lambda = 0$  and  $s = 1$  (the smallest value of  $\lambda$  and  $s$ ); for higher value of  $\lambda$  while the value  $s$  remains the same, these details are smoothed as shown in Fig. 7(c),

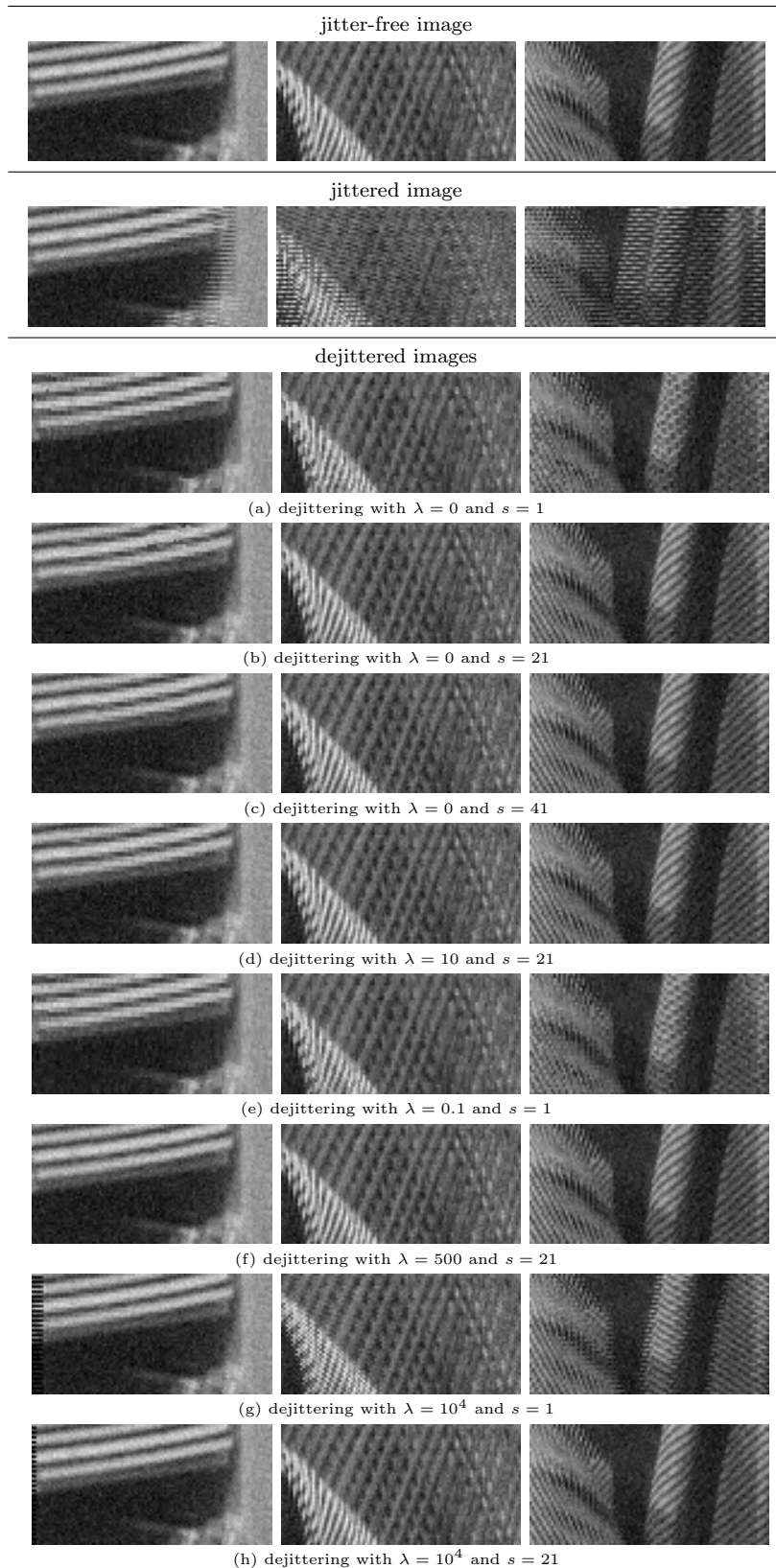


FIGURE 6 – Detail comparison of dejittering result with different parameters of regularization  $\lambda$  and window size  $s$  on Barbara image ( $512 \times 512$ ) corrupted with white Gaussian noise ( $\sigma = 0.04$ ). First row : unjittered image. Second row : jittered image. Third row to ninth row : dejittered images with different values for  $\lambda$  and  $s$ .



(e) and (g). Whereas, the combination of larger  $s$  and  $\lambda = 0$  provides piecewise-constant estimation (also known as staircasing effect) which represents strong discontinuities and is far different from the ground-truth displacement, even though the de-jittered image obtained with this combination (Fig. 5(g)) is very encouraging. This staircasing effect would be reduced by setting a strictly positive value for  $\lambda$  to enforce the smoothness of the estimated displacement (see Fig. 7(d), (f), (h)). In contrast, combining an extremely high value of  $\lambda$  with a small value of  $s$  over-smooths the estimation of the displacement field as depicted in Fig. 7(g) ( $\lambda = 10000$  and  $s = 1$ ); while the same regularization level coupled with large window size (for example,  $\lambda = 10000$  and  $s = 21$ ) produces very similar results to the ground-truth as shown in Fig. 7(h). This behavior reflects how the performance of our method depends on the regularization parameter  $\lambda$  and the size  $s$  of the estimation window. As presented in Fig. 8 which shows the PSNR and SSIM curves depending on  $\lambda$  for several values of  $s$ , these indicators of quality vary according to a common pattern: for a fixed  $s$ , when  $\lambda$  increases, the scores increase, reach a maximum, then decrease and tend towards a limit value. From Fig. 8, it is recommended to assign a high value to  $\lambda$  if the estimation window size  $s$  is large.

For large scale comparison, we provide the SSIM values for all the tested images with different noise levels as well as several parameters of regularization and window size in Table 1. It turns out that the performance of our method varies case by case and depends not only on the image itself but also on the amount of noise. In general, the higher noise level the higher optimal values for  $\lambda$  and  $s$ . To illustrate the performance obtained in case of strong noise, we display in Fig. 9 the three best de-jittering results on Mandrill image ( $512 \times 512$ ) corrupted by a white Gaussian noise with standard deviation  $\sigma = 0.1$ . Even in this case which is combined by strong noise and complex texture, by choosing appropriate values for  $\lambda$  and  $s$ , our method is able to restore fine details and vertical structures as we can observe in the three last columns of Fig. 9 showing eye, hair and beard regions of the Mandrill image.

## 5.2 Evaluation on real fluorescence scanner images

For real data, we evaluated the proposed algorithm on an fluorescence image of 8 tissue microarray cores (see Figs. 10 and 11). This image saved in 16-bit-TIFF format was acquired using the an Innopsys's fluorescence scanner named InnoScan 1100AL at the spatial resolution of  $0.5 \mu m$  per pixel with three excitation wavelengths:  $488 \text{ nm}$ ,  $532 \text{ nm}$  and  $635 \text{ nm}$  which are compatible with cyanine dye fluorophores such as Cy2, Cy3 and Cy5. Under these configurations, the size of the acquired image used for the evaluation is  $9544 \times 4704$  pixels. Such a large image in which the absolute displacement between adjacent lines does not exceed 10 pixels requires huge computational resources for de-jittering task and is not convenient for display purpose. In order to reduce the required computational resource and to produce a satisfactory comparison between images before and after the correction of displacement error, we performed the de-jittering only on non-background regions (i.e. tissue core regions) and display several zooms in details.

We noticed that the vertical rolling effect on the jittered image (see two bottom rows of Figs. 10-11) in the red channel ( $488 \text{ nm}$ ) is visibly stronger than in green ( $532 \text{ nm}$ ) and blue ( $635 \text{ nm}$ ) channels, thus the displacement is not the same on the three channels. Up to our knowledge, this is mainly due to the imperfection of the mechanical system which results in a trajectory difference of the three excitation beams. The de-jittering was therefore performed separately on each channel and the result on the three channels was then merged to produce 3-color image as the original. The values for  $\lambda$  and  $s$  in this experience are fixed as  $\lambda = 500$  and  $s = 11$ .

The de-jittering result obtained with these parameters is reported in Figs. 12-13. We can notice that the jitter has been significantly reduced on the three channels. We observe however some regions

TABLE 1 – Dejittering result on simulated data with different noise levels, window sizes and regularization parameters. The SSIM values are computed using the dejittered images and their corresponding unjittered (eventually noisy) version.

		$\lambda = 0$				$\lambda = 0.1$				$\lambda = 1$			
		$s = 1$	$s = 5$	$s = 21$	$s = 41$	$s = 1$	$s = 5$	$s = 21$	$s = 41$	$s = 1$	$s = 5$	$s = 21$	$s = 41$
SSIM	$\sigma = 0$ (Noise-free)												
	Barbara	0.9088	0.9419	0.9614	0.9533	0.9131	0.9456	0.9614	0.9533	0.9063	0.9518	0.9616	0.9534
	Boat	0.9322	0.9624	0.9851	0.9771	0.9387	0.9683	0.9853	0.9771	0.9399	0.9812	0.9861	0.9772
	Cameraman	0.9585	0.9846	0.9913	0.9851	0.9603	0.9888	0.9913	0.9851	0.9583	0.9930	0.9914	0.9852
	Lena	0.9306	0.9576	0.9798	0.9742	0.9411	0.9621	0.9800	0.9742	0.9537	0.9750	0.9805	0.9743
	Mandrill	0.7780	0.8295	0.9128	0.9302	0.8016	0.8471	0.9136	0.9303	0.8109	0.9024	0.9185	0.9313
	Peppers	0.9018	0.9231	0.9533	0.9569	0.9161	0.9289	0.9537	0.9570	0.9187	0.9480	0.9554	0.9573
	Phantom	0.9918	<b>0.9914</b>	0.9876	0.9868	0.9880	0.9908	0.9876	0.9868	0.9797	0.9888	0.9876	0.9867
	Micro-tubule	0.9439	0.9827	0.9914	0.9911	0.9591	0.9845	0.9914	0.9911	0.9757	0.9888	0.9916	0.9912
	$\sigma = 0.02$												
	Barbara	0.8236	0.8647	0.9236	0.9166	0.8365	0.8722	0.9246	0.9168	0.8265	0.8965	0.9282	0.9174
	Boat	0.8308	0.8629	0.9134	0.9140	0.8443	0.8698	0.9141	0.9144	0.8318	0.8902	0.9178	0.9155
	Cameraman	0.8308	0.8450	0.8827	0.8894	0.8416	0.8525	0.8844	0.8900	0.8312	0.8702	0.8917	0.8926
	Lena	0.8230	0.8515	0.9118	0.9170	0.8409	0.8590	0.9129	0.9171	0.8460	0.8887	0.9166	0.9179
	Mandrill	0.7486	0.7962	0.8962	0.9177	0.7733	0.8127	0.8977	0.9179	0.7734	0.8738	0.9034	0.9190
Peppers	0.8068	0.8275	0.8885	0.9047	0.8275	0.8346	0.8893	0.9049	0.8255	0.8662	0.8947	0.9058	
Phantom	0.7647	0.7620	0.7710	0.7895	0.7791	0.7574	0.7716	0.7895	0.7597	0.7553	0.7749	0.7909	
Micro-tubule	0.8415	0.8842	0.9225	0.9321	0.8609	0.8891	0.9226	0.9321	0.8801	0.9064	0.9249	0.9331	
$\sigma = 0.04$													
Barbara	0.7341	0.7815	0.8700	0.8729	0.7508	0.7911	0.8714	0.8731	0.7383	0.8229	0.8769	0.8745	
Boat	0.7315	0.7604	0.8339	0.8423	0.7485	0.7666	0.8345	0.8423	0.7334	0.7928	0.8404	0.8441	
Cameraman	0.7232	0.7309	0.7851	0.7938	0.7376	0.7343	0.7858	0.7942	0.7272	0.7583	0.7909	0.7967	
Lena	0.7145	0.7407	0.8201	0.8452	0.7370	0.7456	0.8212	0.8455	0.7354	0.7790	0.8267	0.8466	
Mandrill	0.7020	0.7416	0.8628	0.8944	0.7277	0.7558	0.8646	0.8947	0.7193	0.8224	0.8720	0.8961	
Peppers	0.7046	0.7268	0.8051	0.8360	0.7291	0.7335	0.8067	0.8360	0.7261	0.7669	0.8140	0.8377	
Phantom	0.5966	0.5991	0.6238	0.6485	0.6196	0.5973	0.6253	0.6495	0.6117	0.5997	0.6323	0.6522	
Micro-tubule	0.7271	0.7644	0.8352	0.8595	0.7490	0.7715	0.8356	0.8594	0.7535	0.7976	0.8396	0.8607	
$\sigma = 0.1$													
Barbara	0.7341	0.7815	0.8700	0.8729	0.7508	0.7911	0.8714	0.8731	0.7383	0.8229	0.8769	0.8745	
Boat	0.7315	0.7604	0.8339	0.8423	0.7485	0.7666	0.8345	0.8423	0.7334	0.7928	0.8404	0.8441	
Cameraman	0.7232	0.7309	0.7851	0.7938	0.7376	0.7343	0.7858	0.7942	0.7272	0.7583	0.7909	0.7967	
Lena	0.7145	0.7407	0.8201	0.8452	0.7370	0.7456	0.8212	0.8455	0.7354	0.7790	0.8267	0.8466	
Mandrill	0.7020	0.7416	0.8628	0.8944	0.7277	0.7558	0.8646	0.8947	0.7193	0.8224	0.8720	0.8961	
Peppers	0.7046	0.7268	0.8051	0.8360	0.7291	0.7335	0.8067	0.8360	0.7261	0.7669	0.8140	0.8377	
Phantom	0.5966	0.5991	0.6238	0.6485	0.6196	0.5973	0.6253	0.6495	0.6117	0.5997	0.6323	<b>0.6522</b>	
Micro-tubule	0.7271	0.7644	0.8352	0.8595	0.7490	0.7715	0.8356	0.8594	0.7535	0.7976	0.8396	<b>0.8607</b>	
		$\lambda = 10$				$\lambda = 500$				$\lambda = 10000$			
		$s = 1$	$s = 5$	$s = 21$	$s = 41$	$s = 1$	$s = 5$	$s = 21$	$s = 41$	$s = 1$	$s = 5$	$s = 21$	$s = 41$
SSIM	$\sigma = 0$ (Noise-free)												
	Barbara	0.8799	0.9597	0.9628	0.9542	0.8237	0.9549	<b>0.9683</b>	0.9577	0.7831	0.9313	0.9670	0.9663
	Boat	0.9315	<b>0.9916</b>	0.9881	0.9778	0.9008	0.9857	<b>0.9916</b>	0.9806	0.8772	0.9665	0.9859	0.9847
	Cameraman	0.9434	<b>0.9953</b>	0.9915	0.9854	0.9299	0.9846	0.9916	0.9867	0.9182	0.9740	0.9843	0.9872
	Lena	0.9551	0.9864	0.9824	0.9748	0.9363	0.9825	<b>0.9876</b>	0.9773	0.9135	0.9669	0.9838	0.9812
	Mandrill	0.8186	0.9536	0.9294	0.9351	0.8068	0.9678	0.9620	0.9547	0.7894	0.9416	0.9711	<b>0.9771</b>
	Peppers	0.9114	0.9685	0.9607	0.9586	0.8874	0.9650	0.9720	0.9650	0.8696	0.9378	0.9696	<b>0.9737</b>
	Phantom	0.9640	0.9851	0.9873	0.9868	0.9492	0.9734	0.9841	0.9863	0.9472	0.9619	0.9769	0.9851
	Micro-tubule	0.9748	0.9934	0.9924	0.9915	0.9538	0.9914	<b>0.9955</b>	0.9938	0.9334	0.9781	0.9926	0.9937
	$\sigma = 0.02$												
	Barbara	0.7967	0.9357	0.9372	0.9201	0.7232	0.9401	<b>0.9606</b>	0.9353	0.6881	0.8984	0.9588	0.9573
	Boat	0.8152	0.9243	0.9275	0.9192	0.7771	0.9357	0.9650	0.9400	0.7561	0.8987	<b>0.9714</b>	0.9626
	Cameraman	0.8123	0.9019	0.9092	0.8998	0.7967	0.9163	0.9574	0.9311	0.7850	0.8909	<b>0.9652</b>	0.9603
	Lena	0.8472	0.9326	0.9284	0.9210	0.8197	0.9303	<b>0.9669</b>	0.9435	0.7935	0.8911	0.9642	0.9629
	Mandrill	0.7761	0.9393	0.9176	0.9229	0.7607	0.9589	0.9578	0.9472	0.7434	0.9230	0.9685	<b>0.9747</b>
Peppers	0.8119	0.9212	0.9086	0.9100	0.7858	0.9253	0.9500	0.9325	0.7697	0.8798	0.9490	<b>0.9548</b>	
Phantom	0.7382	0.7576	0.7860	0.7983	0.7231	0.7428	0.8041	0.8211	0.7200	0.7396	0.7936	<b>0.8281</b>	
Micro-tubule	0.8754	0.9327	0.9335	0.9356	0.8403	0.9315	<b>0.9594</b>	0.9553	0.8135	0.8974	0.9455	0.9586	
$\sigma = 0.04$													
Barbara	0.7036	0.8852	0.8930	0.8803	0.6409	0.9058	0.9461	0.9137	0.6191	0.8525	<b>0.9512</b>	0.9495	
Boat	0.7054	0.8483	0.8565	0.8511	0.6696	0.8524	0.9115	0.8842	0.6575	0.7894	0.9245	<b>0.9278</b>	
Cameraman	0.7034	0.7949	0.8065	0.8040	0.6863	0.8163	0.8595	0.8452	0.6757	0.7754	<b>0.8790</b>	0.8656	
Lena	0.7198	0.8463	0.8464	0.8534	0.6981	0.8442	0.9069	0.8930	0.6793	0.7819	0.9134	<b>0.9248</b>	
Mandrill	0.7072	0.9121	0.8912	0.9011	0.6866	0.9416	0.9505	0.9302	0.6735	0.8907	0.9641	<b>0.9647</b>	
Peppers	0.7034	0.8374	0.8400	0.8450	0.6840	0.8412	0.9240	0.8932	0.6733	0.7718	0.9218	<b>0.9390</b>	
Phantom	0.5875	0.6074	0.6525	0.6612	0.5706	0.6024	0.7137	0.6939	0.5674	0.5957	0.6734	<b>0.7226</b>	
Micro-tubule	0.7382	0.8390	0.8527	0.8658	0.7004	0.8297	0.8907	0.8968	0.6820	0.7744	0.8843	<b>0.9084</b>	
$\sigma = 0.1$													
Barbara	0.6021	0.7752	0.8108	0.8177	0.5679	0.7888	0.8993	0.8703	0.5603	0.7038	0.9083	<b>0.9241</b>	
Boat	0.6004	0.7289	0.7572	0.7680	0.5827	0.7378	0.8479	0.8176	0.5814	0.6606	0.8599	<b>0.8729</b>	
Cameraman	0.6009	0.6602	0.7040	0.7167	0.5827	0.6666	0.7841	0.7722	0.5763	0.6207	0.8063	<b>0.8139</b>	
Lena	0.6010	0.7265	0.7243	0.7472	0.5806	0.7177	0.8188	0.8005	0.5746	0.6606	0.8136	0.8250	
Mandrill	0.6054	0.7924	0.7899	0.8085	0.5899	0.8158	0.8994	0.8676	0.5873	0.7456	<b>0.9549</b>	0.9377	
Peppers	0.5799	0.6996	0.7186	0.7363	0.5662	0.6881	0.8267	0.8003	0.5610	0.6332	0.8224	<b>0.8626</b>	
Phantom	0.4783	0.4974	0.5336	0.5498	0.4629	0.4919	0.5831	0.5961	0.4604	0.4813	0.5617	0.6138	
Micro-tubule	0.5807	0.6954	0.7028	0.7243	0.5679	0.6609	0.7597	0.7715	0.5653	0.6095	0.7193	0.7721	

where the jitter was not well corrected specially in transitive zones between bright and dark features, but in overall the visual effect is sharply improved. Furthermore, in spite of the separate processing on each channel, the merged image of these dejittered versions does not suffer from decorrelation artifact, thus no registration is required to realign the dejittering result.

An example of dejittering result on image acquired in full-width-scanning is also illustrated in Fig. 3 showing a tool slide with checkerboard pattern used for calibrating scanners. The width of this image which is scanned at  $1.0 \mu m$  per pixel exceeds 20 thousand pixels corresponding to more than 20 millimeters in real scale. Since the distance between two vertical lines of the images can reach several hundreds of pixels, large estimation window (high value of  $s$ ) is considered. As shown in the two bottom rows of Fig. 3, we noticed that the result obtained without regularization is very similar to those with regularization in terms of visual quality. However, the computing time of the former is much faster than the latter (few minutes compared with more than one hour).

In general, the computational cost depends on the size of the input images as well as the parameter setting. The computing time is in average  $3.8 \pm 0.7$  seconds in the case of dejittering without regularization and  $26.9 \pm 9.8$  seconds with regularization on images of size  $512 \times 512$  pixels. On real large images whose the width can reach several thousands of pixels as shown in this section, the computing time varies from less than 5 minutes to more than 10 minutes for dejittering without regularization and it is about more than one hour with regularization. The experiments were performed on a Macbook Pro equipped with 2.7 Ghz Intel Core i7, 16 Gb of RAM and the Mac OS X v. 10.12.4 operating system. The algorithm was implemented in C/C++ and we exploited the intrinsic parallelism of the CPU to solve large-scale problems.

## 6 Conclusion

In this paper, we have proposed a variational method to remove a specific jitter arising on large fluorescence scanner images. We reformulate the dejittering problem as an estimation problem of the underlying displacement by minimizing an energy functional which combines convex regularization and non-convex non-local data fidelity term. To solve the non-convex minimization problem, the half-quadratic splitting technique is used to decompose the energy functional into convex and non-convex parts in such a way that the original minimization problem can be done by alternating two optimization steps. One of which uses the projected gradient algorithm for the convex part and the other uses an exhaustive search for the non-convex parts. This approach allows to approximate accurately the optimal solution of the original problem while avoiding the linearization of the data term. In numerous experiments, we show that our method is able not only to remove efficiently the rolling effect due to this jitter even in the case of huge images and large, non-integer displacements.

## Références

- [1] Anil C. Kokaram. Line registration for jittered video. In *Motion Picture Restoration : Digital Algorithms for Artefact Suppression in Degraded Motion Picture Film and Video*, pages 99–118. Springer London, London, 1998.
- [2] A. Kokaram, P. Rayner, P. van Roosmalen, and J. Biemond. Line registration of jittered video. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97)*, volume 4, pages 2553–2556, 1997.

- [3] L. Laborelli. Removal of video line jitter using a dynamic programming approach. In *Proc. IEEE International Conference on Image Processing (ICIP 2003)*, volume 2, pages II-331-4 vol.3, 2003.
- [4] Jianhong Shen. Bayesian video dejittering by the BV image model. *SIAM Journal on Applied Mathematics*, 64(5) :1691-1708, 2004.
- [5] Sung Ha Kang and Jianhong Shen. Video dejittering by bake and shake. *Image and Vision Computing*, 24(2) :143-152, 2006.
- [6] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7) :629-639, 1990.
- [7] Sung Ha Kang and Jackie Shen. Image dejittering based on slicing moments. In Xue-Cheng Tai, Knut-Andreas Lie, Tony F. Chan, and Stanley Osher, editors, *Image Processing Based on Partial Differential Equations : Proceedings of the International Conference on PDE-Based Image Processing and Related Inverse Problems, CMA, Oslo, August 8-12, 2005*, pages 35-55. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [8] Mila Nikolova. One-iteration dejittering of digital video images. *Journal of Visual Communication and Image Representation*, (4) :254-274, 2009.
- [9] Mila Nikolova. Fast dejittering for digital video frames. In Xue-Cheng Tai, Knut Mørken, Marius Lysaker, and Knut-Andreas Lie, editors, *Scale Space and Variational Methods in Computer Vision : Second International Conference, SSVM 2009, Voss, Norway, June 1-5, 2009. Proceedings*, pages 439-451. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [10] Frank Lenzen and Otmar Scherzer. Partial differential equations for zooming, deinterlacing and dejittering. *International Journal of Computer Vision*, 92(2) :162-176, 2011.
- [11] Gerhard Huisken. Flow by mean curvature of convex surfaces into spheres. *Journal of Differential Geometry*, 20(1) :237-266, 1984.
- [12] Francine Catté, Pierre-Louis Lions, Jean-Michel Morel, and Tomeu Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical Analysis*, 29(1) :182-193, 1992.
- [13] Guozhi Dong, Aniello Raffaele Patrone, Otmar Scherzer, and Ozan Öktem. Infinite dimensional optimization models and pdes for dejittering. In Jean-François Aujol, Mila Nikolova, and Nicolas Papadakis, editors, *Scale Space and Variational Methods in Computer Vision : 5th International Conference, SSVM 2015, Lège-Cap Ferret, France, May 31 - June 4, 2015, Proceedings*, pages 678-689. Springer International Publishing, Cham, 2015.
- [14] Matthias Ghodstinat, Andrés Bruhn, and Joachim Weickert. Deinterlacing with motion-compensated anisotropic diffusion. In Daniel Cremers, Bodo Rosenhahn, Alan L. Yuille, and Frank R. Schmidt, editors, *Statistical and Geometrical Approaches to Visual Motion Analysis : International Dagstuhl Seminar, Dagstuhl Castle, Germany, July 13-18, 2008. Revised Papers*, pages 91-106. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [15] Sune Keller, François Lauze, and Mads Nielsen. A total variation motion adaptive deinterlacing scheme. In Ron Kimmel, Nir A. Sochen, and Joachim Weickert, editors, *Scale Space and PDE Methods in Computer Vision : 5th International Conference, Scale-Space 2005, Hofgeismar, Germany, April 7-9, 2005. Proceedings*, pages 408-418. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [16] S. H. Keller, F. Lauze, and M. Nielsen. Deinterlacing using variational methods. *IEEE Transactions on Image Processing*, 17(11) :2015-2028, 2008.

- [17] Andrés Almansa, Vicent Caselles, Gloria Haro, and Bernard Rougé. Restoration and Zoom of Irregularly Sampled, Blurred, and Noisy Images by Accurate Total Variation Minimization with Local Constraints. *SIAM Multiscale Modeling & Simulation*, 5(1) :235–272, Jan 2006.
- [18] Gabriele Facciolo, Andrés Almansa, Jean-François Aujol, and Vicent Caselles. Irregular to Regular Sampling, Denoising, and Deconvolution. *SIAM Multiscale Modeling & Simulation*, 7(4) :1574–1608, Jan 2009.
- [19] Thomas Brox and Jitendra Malik. Large displacement optical flow : Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3) :500–513, 2011.
- [20] Denis Fortun, Patrick Bouthemy, and Charles Kervrann. Optical flow modeling and computation : A survey. *Computer Vision and Image Understanding*, 134 :1–21, 2015.
- [21] Bertrand Mercier. *Lectures on Topics in Finite Element Solution of Elliptic Problems*. Springer-Verlag Berlin Heidelberg, 1979.
- [22] Jonathan Eckstein and Dimitri P. Bertsekas. On the douglas–rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1) :293–318, 1992.
- [23] Patrick L. Combettes. Solving monotone inclusions via compositions of nonexpansive averaged operators. *Optimization*, 53 :475–504, 2004.
- [24] Patrick L. Combettes and Jean-Christophe Pesquet. *Proximal Splitting Methods in Signal Processing*, pages 185–212. Springer New York, New York, NY, 2011.
- [25] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1) :120–145, 2011.
- [26] Laurent Condat. A generic proximal algorithm for convex optimization – application to total variation minimization. *IEEE Signal Processing Letters*, 21(8) :985–989, 2014.
- [27] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1) :185–203, 1981.
- [28] Christopher Zach, Thomas Pock, and Horst Bischof. *A Duality Based Approach for Realtime TV-L<sup>1</sup> Optical Flow*, pages 214–223. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [29] Andreas Wedel, Thomas Pock, Christopher Zach, Horst Bischof, and Daniel Cremers. *An Improved Algorithm for TV-L<sup>1</sup> Optical Flow*, pages 23–45. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [30] F. Steinbrücker, T. Pock, and D. Cremers. Large displacement optical flow computation without warping. In *Proc. IEEE 12th International Conference on Computer Vision (ICCV 2009)*, pages 1609–1614, 2009.
- [31] D. Geman and Chengda Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing*, 4(7) :932–946, 1995.
- [32] Patrick L. Combettes and Valérie R. Wajs. Signal recovery by proximal forward-backward splitting. *SIAM Multiscale Modeling & Simulation*, 4(4) :1168–1200, 2005.

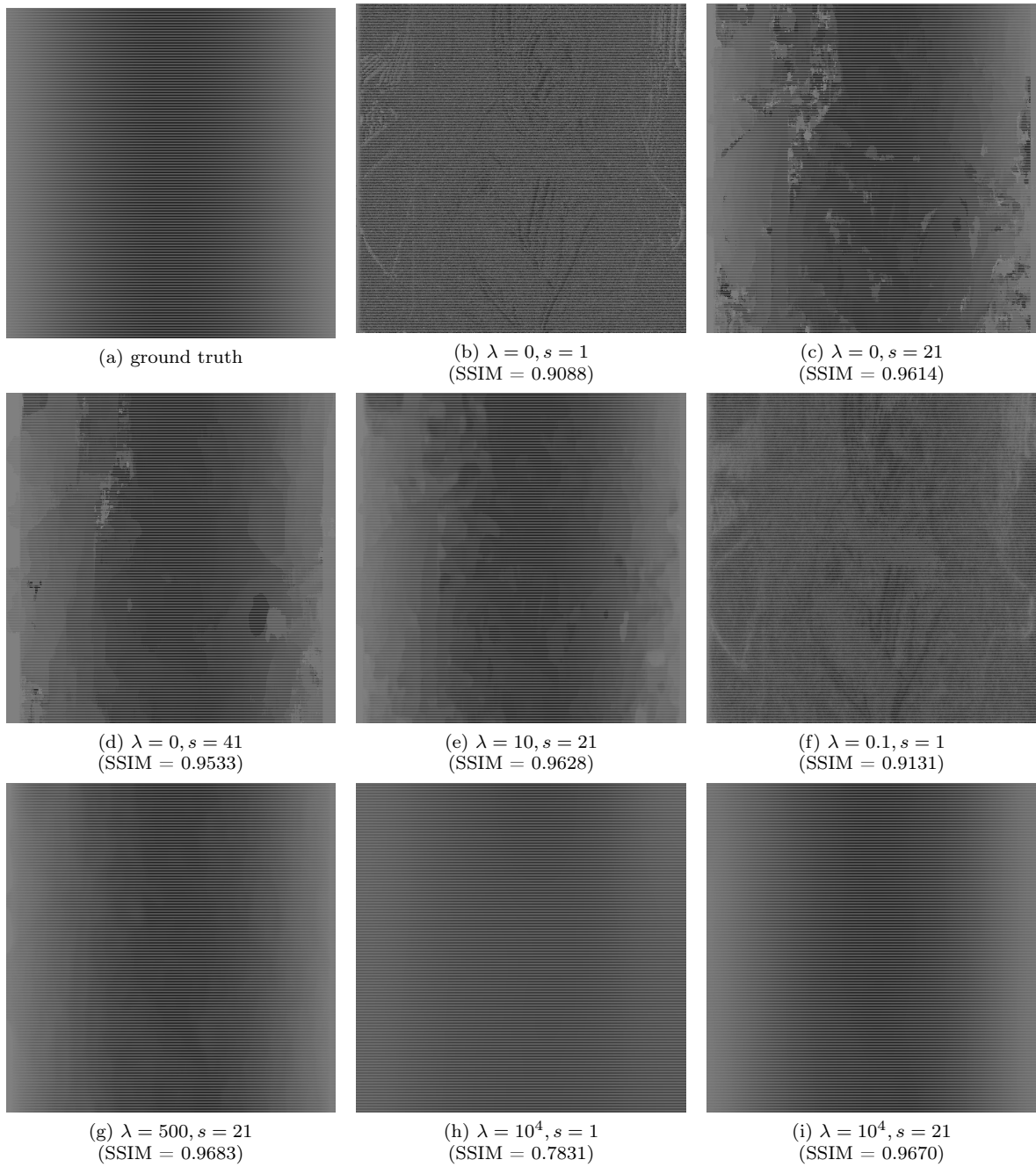


FIGURE 7 – Comparison of estimated displacement field with different parameters of regularization  $\lambda$  and window size  $s$  on Barbara image ( $512 \times 512$ ) corrupted by white Gaussian noise  $\sigma = 0.04$ . Top left : Ground-truth. (a)-(h) : Displacement estimated with different values for  $\lambda$  and  $s$ .

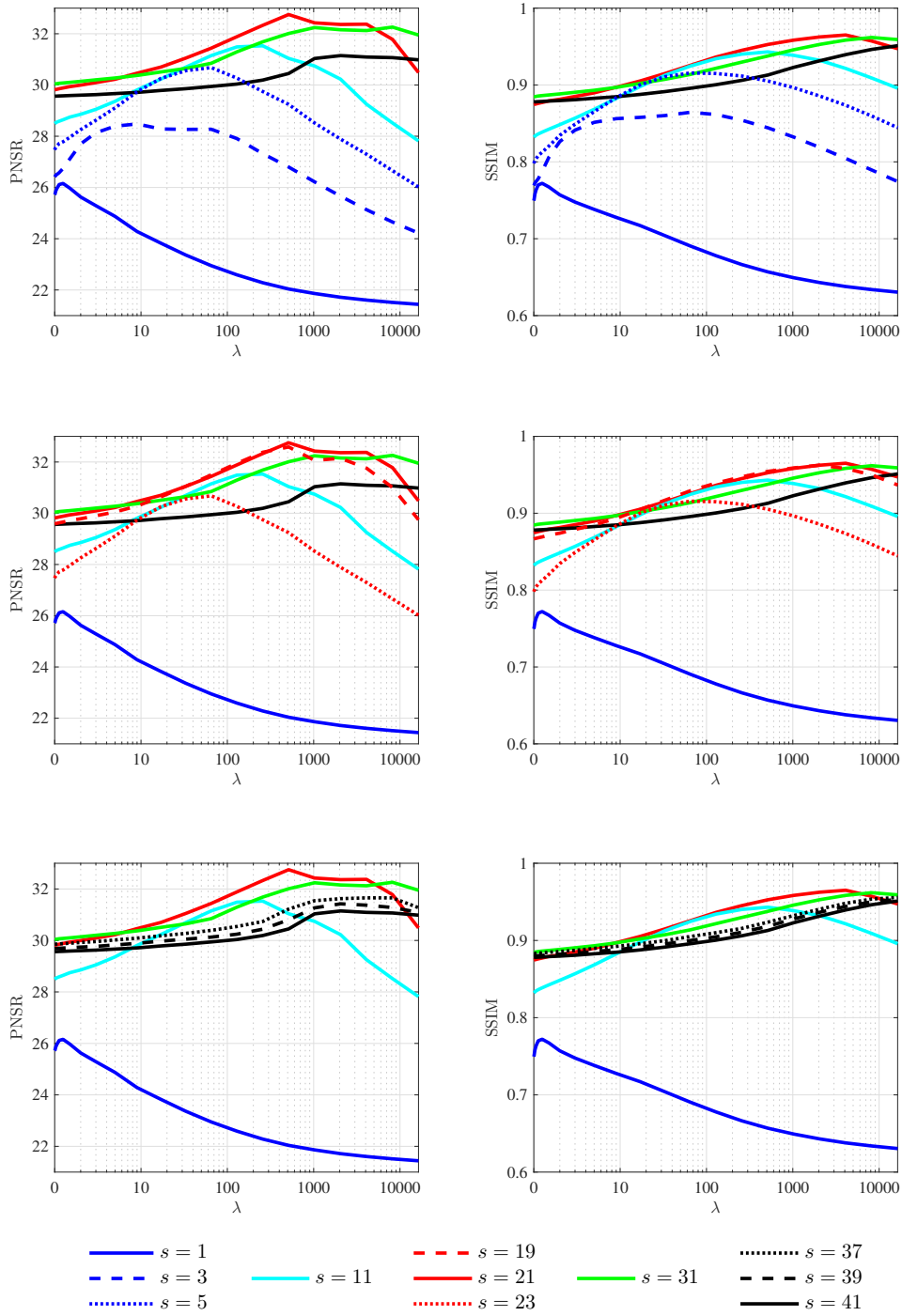


FIGURE 8 – **Dejittering performance (PSNR and SSIM values) depending on  $\lambda$  for several values of  $s$ .** The scores are computed on Barbara image ( $512 \times 512$ ) corrupted by Gaussian white noise ( $\sigma = 0.04$ ) and are plotted in logarithmic scale for comparison. We display the curves corresponding to three categories of window size : small window (blue curves), medium window (red curves) and large window (black curves). The cyan and the green curves which correspond to  $s = 11$  and  $s = 31$  respectively are transitional between these groups.

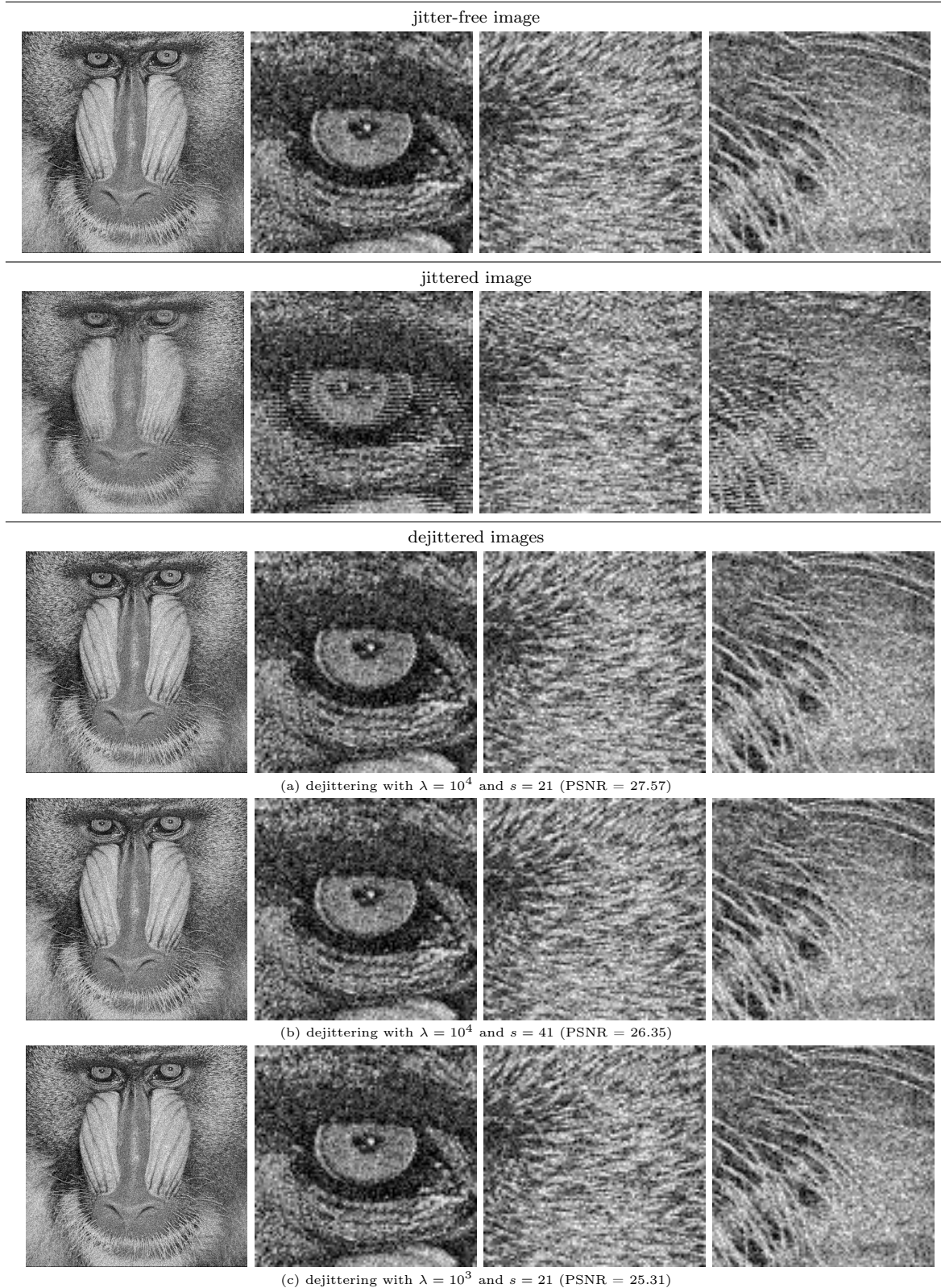


FIGURE 9 – **Dejittering result on very noisy Mandrill image ( $512 \times 512$ ) (corrupted with white Gaussian noise  $\sigma = 0.1$ ).** The first column contains full size images, all remaining columns display some regions of interest to compare in details. First row : jitter-free image. Second row : jittered image. Third row to fifth row : dejittering with different values for  $\lambda$  and  $s$ .



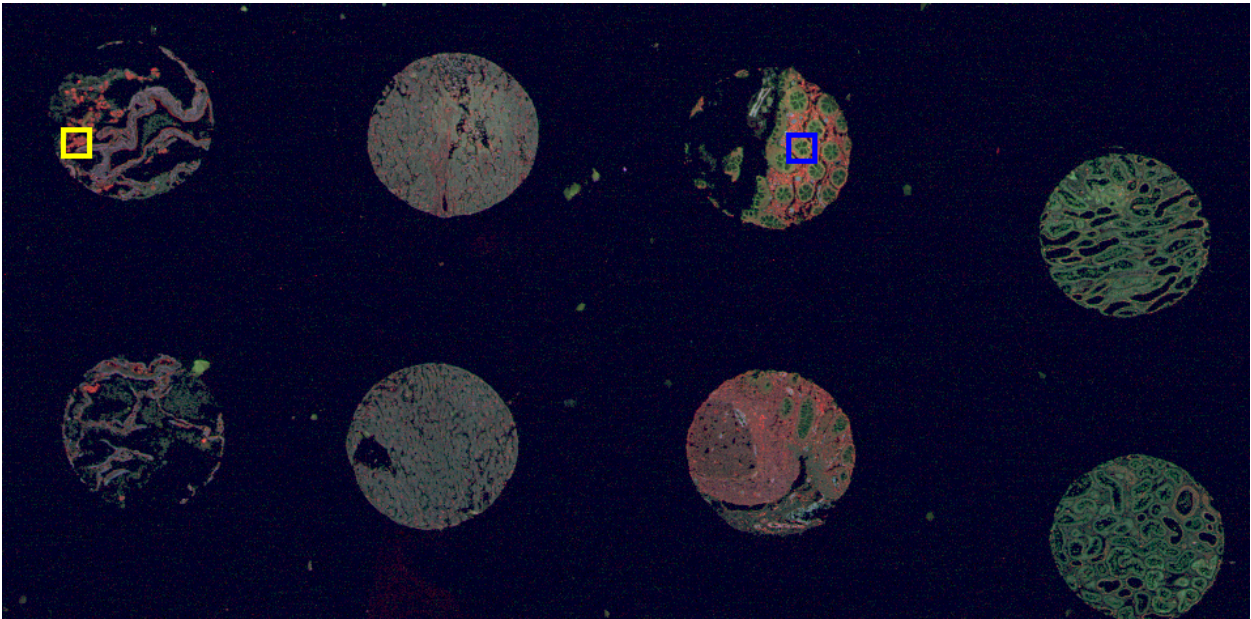


FIGURE 10 – Jittered image full size image ( $9544 \times 4704$  pixels) of 8 tissue microarray spots acquired in 3 colors with resolution of  $0.5 \mu\text{m}/\text{pixel}$ .

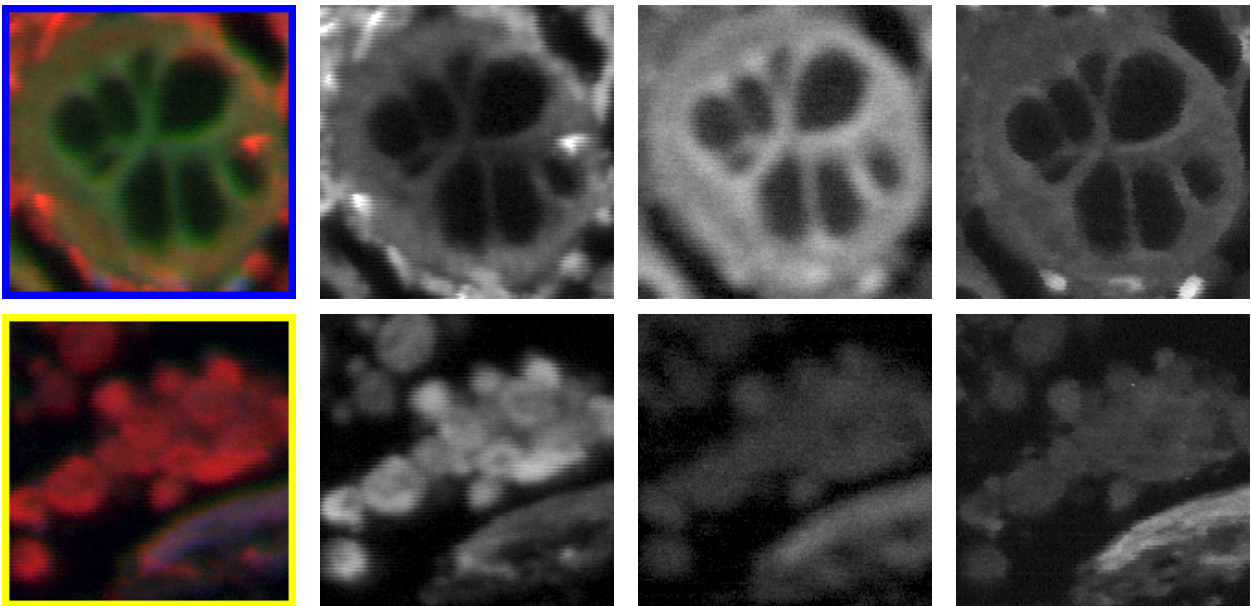


FIGURE 11 – Jittered image of 8 tissue microarray spots acquired in 3 colors with resolution of  $0.5 \mu\text{m}/\text{pixel}$ . Zoom-in view of two selected areas (blue box and yellow box respectively) in Fig. 10; from left to right : 3 colors at the same time, red ( $488 \text{ nm}$ ), green ( $532 \text{ nm}$ ) and blue ( $635 \text{ nm}$ ) channels displayed separately.

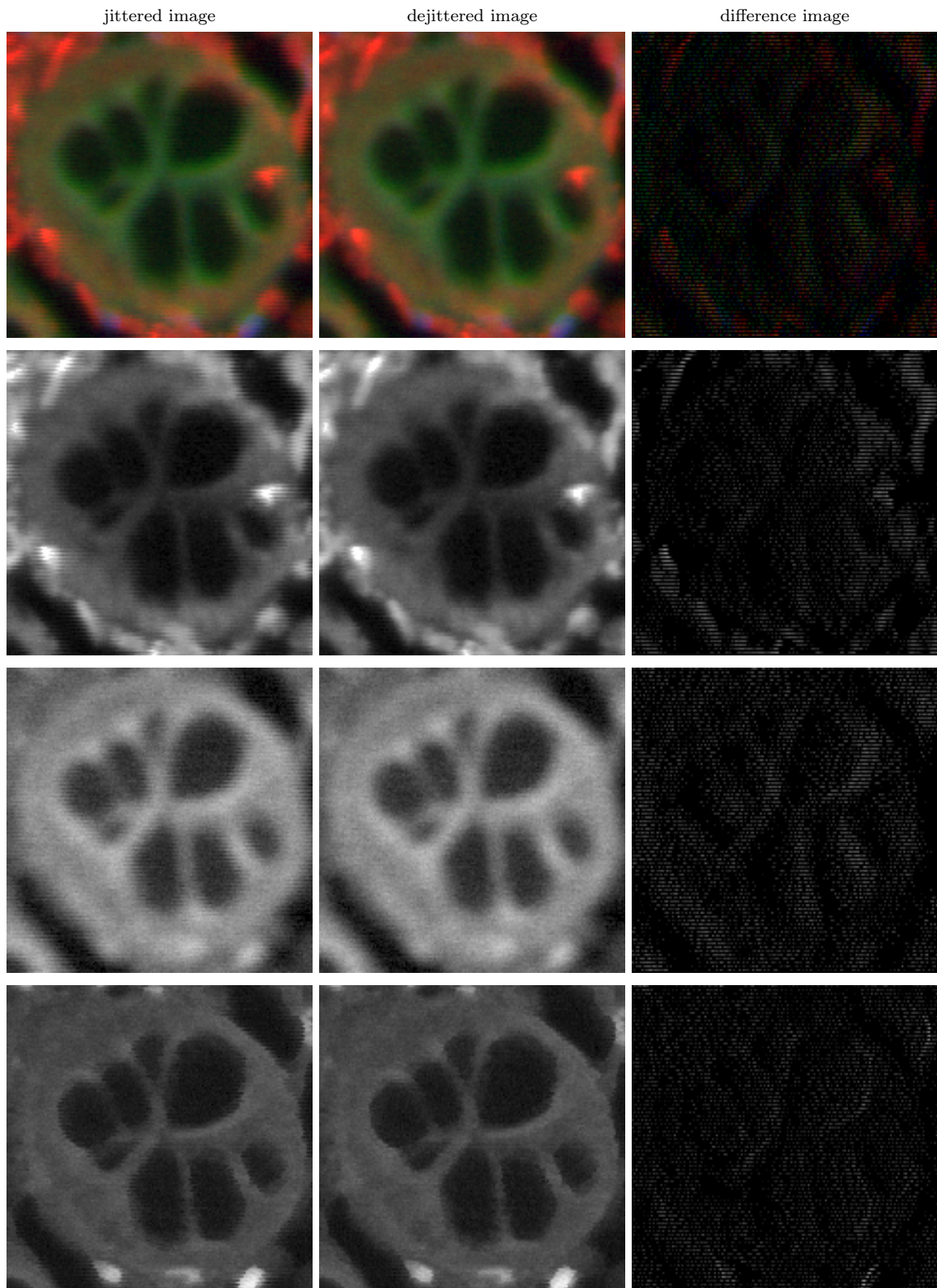


FIGURE 12 – Dejittering result on TMA image in Fig. 10 (area inside of the blue box). From top to bottom : 3-color image, red (488  $nm$ ), green (532  $nm$ ) and blue (635  $nm$ ) channel. From left to right : jittered image, dejittered image and difference between them.

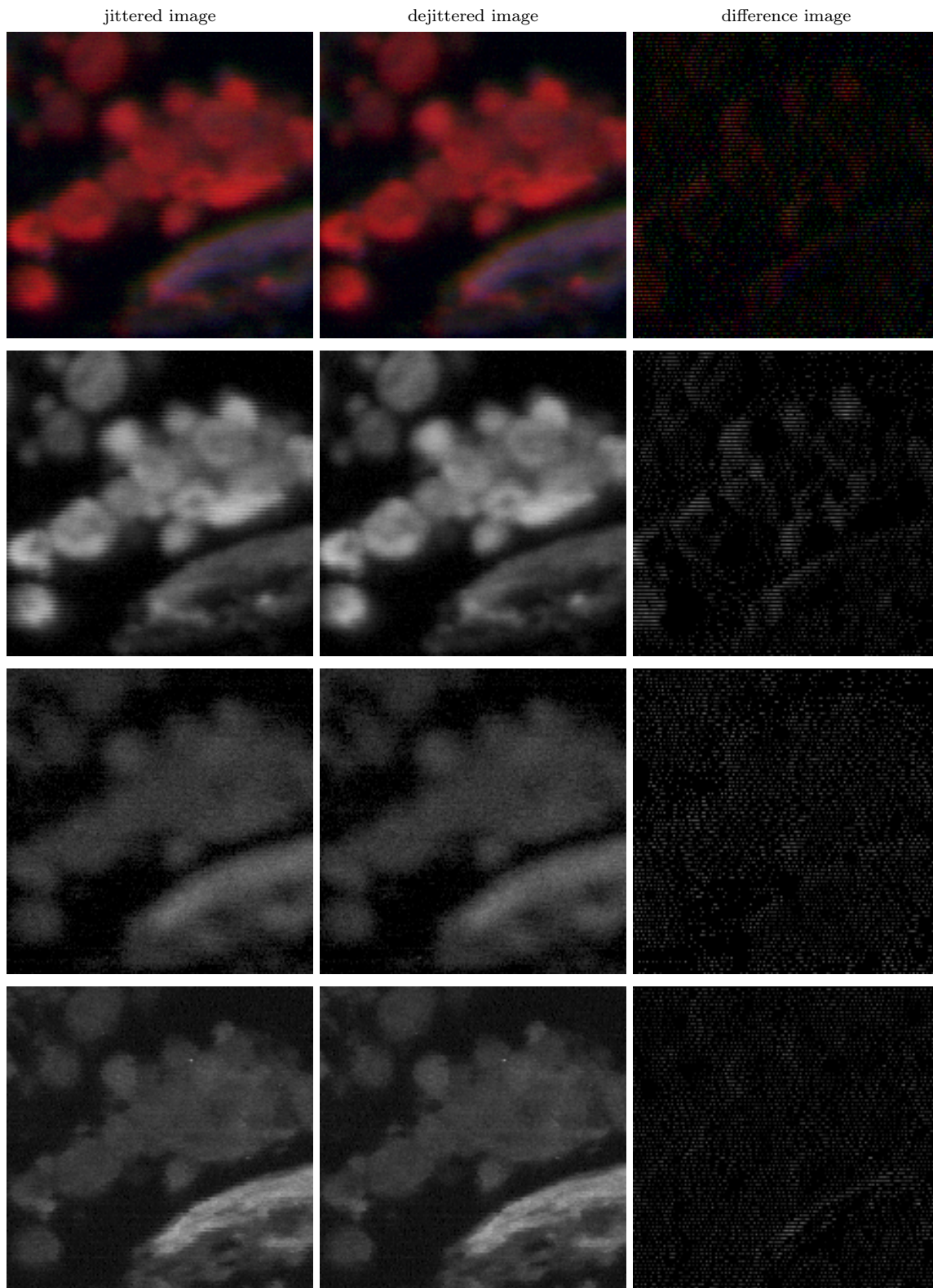


FIGURE 13 – **Dejittering result on TMA image in Fig. 10 (area inside of the yellow box).** From top to bottom : 3-color image, red (488 *nm*), green (532 *nm*) and blue (635 *nm*) channel. From left to right : jittered image, dejittered image and difference between them.