



Efficient segmentation and positioning of 3D fluorescent spherical beads in confocal microscopy

Alessandro Benfenati, Francesco Bonacci, Tarik Bourouina, Hugues Talbot

► To cite this version:

Alessandro Benfenati, Francesco Bonacci, Tarik Bourouina, Hugues Talbot. Efficient segmentation and positioning of 3D fluorescent spherical beads in confocal microscopy. *Journal of Mathematical Imaging and Vision*, 2021, 63 (1), pp.56-72. 10.1007/s10851-020-00994-1 . hal-02150316v1

HAL Id: hal-02150316

<https://hal.science/hal-02150316v1>

Submitted on 7 Jun 2019 (v1), last revised 19 Feb 2021 (v5)

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.

Efficient segmentation and positioning of 3D fluorescent spherical beads in confocal microscopy^{*}

Alessandro Benfenati^{a,*}, Francesco Bonacci^b, Tarik Bourouina^c, Hugues Talbot^d

^a*Dipartimento di Scienze e Politiche Ambientali, Via Celoria 2, 20133, Milano, Italy*

^b*IFSTTAR, ENPC, Université Paris-Est, 14-20 Boulevard Newton, 77420, Champs sur Marne, France*

^c*ESYCOM, ESIEE, ENPC, Université Paris-Est, 5 Boulevard Descartes 77420 Champs-sur-Marne, France*

^d*CVN, Inria, CentraleSupélec, Université Paris-Saclay, 9 Rue Joliot Curie, 91190, Gif-sur-Yvette, France*

Abstract

Particle estimation is a classical problem arising in many science fields, such as biophysics, fluid mechanics, bio-medical imaging. Many interesting applications in these areas involve 3D data: this work presents a technique to estimate the 3D coordinates of the center of spherical particles. This procedure provides an estimation of both the center and the profile of the 2D intersections of the particles with the frames, by coupling the usage of Total Variation functional and of a regularized weighted Least Square fit. The 2D information is used to retrieve the 3D coordinates using geometrical properties. The performance of this procedure strongly depends on the quality of the acquisition, hence a particularly tailored denoising technique is applied for Poisson noise: this leads to a better estimation of the particle positions.

Keywords: Particle estimation, particle tracking, 3D data, Brownian Motion

^{*}Funding for this project was provided in part by LABEX MMCD and ANR CoMeDIC.

^{*}Corresponding author

Email addresses: alessandro.benfenati@unimi.it (Alessandro Benfenati), francesco.bonacci@ifsttar.fr (Francesco Bonacci), tarik.bourouina@esiee.fr (Tarik Bourouina), hugues.talbot@centralesupelec.fr (Hugues Talbot)

1. Introduction

Particle tracking techniques are widely employed in several science fields for identifying particular structures or processes of interest. Some important examples include biophysics, where these techniques are involved in the observation of molecular level motion of kinesin in microtubules and of motion of myosin on actin [1, 2], in the study of the infection path of a virus [3, 4] or in the investigation of cytoskeletal filaments [5]; another topic involving particles tracking problem regards the observation of protein motion in cell membranes [6, 7, 8] or intracellular transport [9]. Other interesting areas of application include fluid dynamics and mechanics, in particular Rheology [10], where the thermal motion of Brownian particles has been tracked to study local rheological properties [11, 12]; complex fluids [13, 14]; and Microrheology in Medicine [15]. Colloidal works have benefited from developments in particle tracking procedures in the study of phase transition [16] and of pair interaction potential [17]. It has been pointed out [18] that *particles* have different meanings depending on the applications: a single molecule, a virus, a spherical object. In this work, a particle is a spherical object around 1 micrometer in diameter, observed in confocal microscopy.

Particle tracking consists of two main steps: particle position estimation and trajectory reconstruction. Following [19], in this work a numerical simulation of the standard setup is adopted: the simulated system consists of a CCD camera connected to a microscope which records images (frames) of molecules or spherical particles. In [19] (and references therein), a particular focus on microrheology-related problems is considered, and the balance between high spatial resolution and timescale of data acquisition is considered in depth: the former leads to approximate multiple-tracking techniques while the latter allows a greater flexibility and provides an high statistical accuracy. In [20] the spatial resolution influence was investigated. In the presented paper, the first step of particle tracking problem is solved: the proposed algorithm provides estimations of the particles position with subpixel resolution, both in two and

three dimensional cases. The proposed procedure aims mainly to treat the *static error* [19], which arise from noise affecting this type of experiments; this *static error* is equivalent to the notion of *precision* in [20].

In the past, several procedure have been proposed to estimate the particle position: cross-correlation of a sequence of images [1, 21, 22], centroid techniques [23, 24, 25, 26], Gaussian fitting [27, 28]. Some of them claim subpixel resolution, and in [20] a wide comparison of these techniques showed that significant numerical experimentation is needed before validating such results. Other methods includes combinatorial optimization [29], nearest neighbour [30], Kalman filtering coupled with probabilistic data association [31], use of the Viterbi algorithm [32] and several others. An experimental comparison of a plethora of methods can be found in [18]. In this work, our proposed procedure is first tested on synthetic but realistic data. The algorithm proved itself to be providing good performance on such data, hence it is applied on real 3D data with satisfactory results.

The procedure presented in this paper provides position estimations of 3D spherical particles: this approximation is inspired by the problem of estimating the motion of spherical nanoparticles suspended in a fluid. A novel approach based on Total Variation functional and on Least Square fitting is proposed to locate the center of the spherical particles in 2D frames. The 3D centers of the particles are hence estimated using geometric properties and employing the 2D information retrieved in the previous steps. The algorithm achieves subpixel resolution both in the 2D case, i.e. in estimating the position of the particles within frames, and in the 3D case.

This paper is organized as follows: in Section 2 the simulation procedure is described, in order to get realistic 3D data to validate the proposed algorithm. In Section 3 details of the proposed procedure are given: the pre-processing of the frames and the estimation of the 2D centers, and then the 3D estimation. Section 4 is devoted to the numerical experimentation on both synthetic and realistic data; finally, in Section 5, conclusions are drawn.

Notation. Bold letters, bold capital letters and Latin (or Greek) letters

denote vectors, matrices and scalars, respectively. The i -th element of the vector \mathbf{x} is denoted by x_i . The notation $\mathcal{N}(\mu, \sigma^2)$ indicates a Gaussian distribution of mean μ and variance σ^2 . \mathbf{I} denotes the identity matrix, $\mathbf{0}$ the vector with all
65 zeros entries.

2. Data Creation: Simulation Procedure

The synthetic datasets used to validate the proposed algorithm are simulated following these steps, which are inspired by the characteristics of real settings:

- N spherical particles of radius a are randomly placed in a 3D volume of
70 dimension $D_x \times D_y \times D_z$. The particles are assumed to have all the same, known radius;
- the 3D volume is discretized into an array of $N_x \times N_y \times N_z$ voxels; each voxel has dimension $dx \times dy \times dz$, being $dx = D_x/N_x, dy = D_y/N_y, dz = D_z/N_z$. N_z represents the number of 2D frames. Each particle is dis-
75 cretized in this volume;
- aiming to simulate realistic data, a blurring operator is applied to each frame, then Gaussian and/or Poisson noise is respectively added to or composed with each image.

In the following the creation of the dataset is described precisely

80 *Position simulation.* The continuous positions $\{\mathbf{x}_i\}_{i=1,\dots,N}$ of the N particles are randomly chosen in $D_x \times D_y \times D_z$, via an uniform distribution. The 3D position of the i -th particle is denoted via $\mathbf{x}_i = (x_i, y_i, z_i)^\top$.

Discretization. Given the continuous coordinates \mathbf{x}_i of the i -th particle and the radius a , the voxels at distance less or equal to a are filled with a value of H ,
85 while the others are set to h , aiming to have a non-zero constant background. In our simulations, we set $h = 10$ and $H = 220$. These values were chosen in order to simulate realistic `tiff` images, which usually have values in $[0, 255]$. In

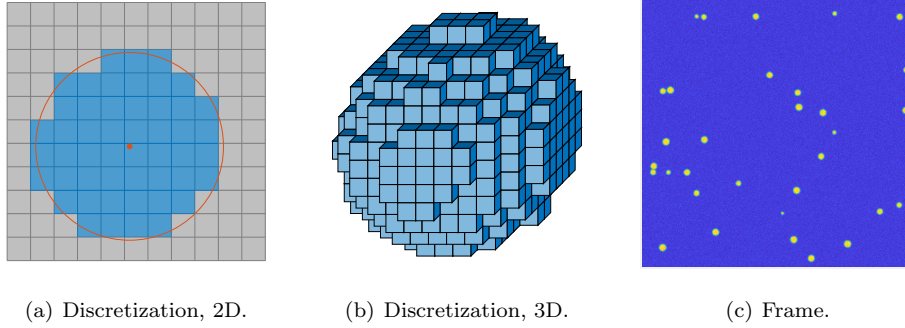


Figure 1: Panel 1(a): discretization of a disk. The true center is represented by the orange dot together with the true profile in the same color. The pixels at a distance less than a are set to H (highlighted in light blue), while the others are set to h . It is clear that it is not always possible to discretize the disk in a symmetric fashion. The procedure follows the same ratio for the 3D case (1(b)). Panel 1(c): blurred and noisy frame

Figure 1 a 2D explanation of this procedure is depicted: the 3D case follows the same procedure.

Blurring and Noise. A blurring operator of Gaussian type (dimension: 5×5 pixels, of zero mean and unitary variance, created via the MatLab function `imfilter`) is applied to each frame, simulating the perturbation given by the acquisition system. Gaussian noise of level σ_n is added to each frame: let be $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \sigma_n \mathbf{I})$ a realization of a Gaussian multivalued random variable of zero mean and covariance matrix $\sigma_n \mathbf{I}$. The noise $\boldsymbol{\eta}$ is added according to the following formula (which is a slight modification of the one in [33])

$$\mathbf{F}_z = \mathbf{F}_z + \sigma_n \frac{\boldsymbol{\eta}}{\|\boldsymbol{\eta}\|_F} (1 + \|\mathbf{F}_z\|_F)$$

90 being \mathbf{F}_z the z -th frame and $\|\cdot\|_F$ the Froebenius norm. A different $\boldsymbol{\eta}$ is created for each frame. Moreover, in order to have the most realistic data, Poisson noise is composed with the images, via the MatLab function `imnoise`, employed by the rescaling `1e12*imnoise(1e-12*F,'Poisson')`, being \mathbf{F} the current frame (see the MatLab help for the `imnoise` function for more details about this
95 procedure.). Finally, the intensity values of each frame are rescaled into the

interval $[0, 255]$. See Figure 1(c) for a visual inspection of the result.

3. Algorithm

The steps for the particles recognition problem in the 3-dimensional case are presented in Algorithm 1:

Algorithm 1 Let N_z be the frames's number, a the radius of the particles.

- 1: **for** $z = 1, \dots, N_z$ **do**
 - 2: Denoising of z -th frame.
 - 3: Search for the K connected components $\{L_k\}_{k=1, \dots, K}$, in the z -th frame.
 - 4: **for** $k = 1 \dots, K$ **do**
 - 5: Compute the center of mass m_k of the k -th component.
 - 6: Open a window in the denoised frame, centered in m_k .
 - 7: Compute the k -th center via a regularized weighted Least Square fit.
 - 8: Create the two candidates for computing the center of the particle in 3D.
 - 9: Compute the estimated centers of the particles via a weighted mean.
-

100 Subsection 3.1 is devoted to illustrating the idea and the procedures beyond lines 2–7 of Algorithm 1, while Subsection 3.2 explains how the 2D information obtained from the frames can be used to estimate the particle center coordinates in 3 dimensions.

3.1. Frames Processing

105 The procedures described in lines 2–7 are listed and expanded below.

Denoising. The presence of noise, together with the blurring operator, could lead to some artefacts in the particle position and diameter estimation, hence a denoising and deblurring procedure is necessary. A simple approach is using a Gaussian filtering [34]: this procedure is very quick and inexpensive, performed
110 via the FFT MatLab's native algorithms, see Figure 2(b) for the results. The pros of this approach are that it reduces the presence of the noise and in its speed; while the drawbacks lie in the fact that the image is oversmoothed: the perturbing effect of the PSF is augmented , resulting in blurred edges.

We propose a method based on an optimization method: given the noisy and blurred frame \mathbf{g} , one is led to compute the denoised frame $\tilde{\mathbf{f}}$ as

$$\tilde{\mathbf{f}} = \underset{\mathbf{f} \in \mathcal{C}}{\operatorname{argmin}} f_0(\mathbf{H}\mathbf{f} + b; \mathbf{g}) + \mu f_1(\mathbf{f})$$

where \mathcal{C} is a convex, non-empty closed set of constrains (e.g., the non-negative orthant), \mathbf{H} is the blurring operator representing the PSF, b is a constant back-ground term, $\mu > 0$ is a real parameter and f_0 and f_1 are the *fit-to-data* and *regularization* functions, respectively. This problem has been deeply investigated in recent years, leading to the development of a great number of valid optimization algorithms [35, 36, 37, 38]. Moreover, this formulation of the problem allows us to choose the function f_1 in order to preserve some desired characteristic (e.g., sharp edges) on the recovered image.

Search for the connected components. In order to get an estimation of the profile and of the center of the particles in the current frame, they must be localized first. The strategy is quite simple: the first step consists of thresholding the denoised frame, by employing the Otsu method [39, 34] (see Figure 2(c)). Then, the K connected components $\{L_k\}_{k=1,\dots,K}$ in the thresholded frame are recognized and labeled (Figure 2(d)). The Matlab function `bwlabel` is set to assume the 8-connected neighbours. At this stage, the area of each k -th connected component is stored in a_k : this area will be used for the estimation in 3 dimensions of the center (see (3)). The center of mass m_k of L_k is computed, together with a first raw estimation r_k of the radius: r_k is the distance of m_k from the furthest pixel in L_k (Figure 3(a)).

Least Square Fit. Once the connected components are recognized, a least square fit is performed on each one in order to estimate the profile and the center of the particle. First of all, a Total Variation functional [37] is applied to the current denoised frame, namely \mathbf{D} , aiming to find the edges of the particles (Figure 3(c)). Denoting (with an abuse of notation) the partial derivatives via ∂_x and ∂_y in the two directions, the Total Variation function on \mathbf{D} reads as

$$\mathbf{TV}(\mathbf{D}) = \sqrt{(\partial_x \mathbf{D})^2 + (\partial_y \mathbf{D})^2}. \quad (1)$$

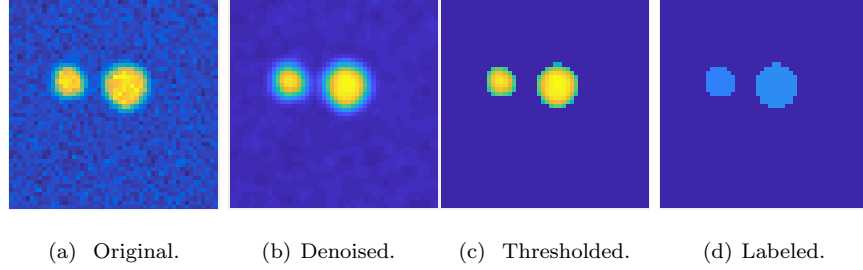


Figure 2: Particular of the frame of Figure 1(c). From left to right: a region of interest with two separated particles. Second panel: result of the Gaussian filtering. The noise is reduced, but the edges are blurred. Third panel: thresholding via the Otsu method. Last panel: labeling procedure, where different colors mean different labels. The order of labeling does not influence the final result.

The data are discrete, hence a discrete version of \mathbf{TV} is implemented: the derivatives are computed via centered differences with 2nd order accuracy. Centered
135 differences with 4th order accuracy were tested, but no significant differences were observed in the final results.

For sake of clarity, we focus on the k -th component, assuming that is well separated from all the others.

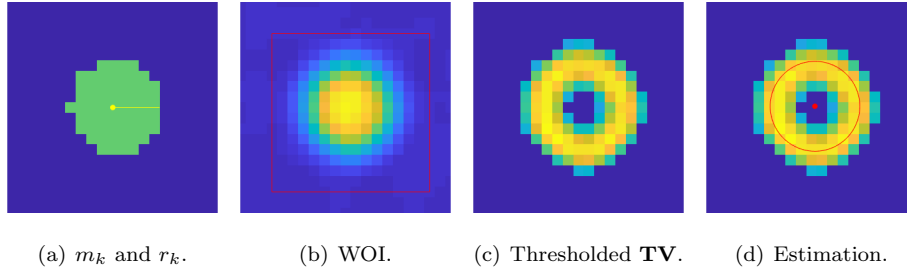


Figure 3: Procedure for the least square fit, focusing on a single connected component. First panel: connected component, with its center of mass and raw radius estimation. Second panel: window of interest around the localized particle. Third panel: chosen pixels for the least square fit, with the relative intensity values. Fourth panel: estimated center together with the profile, based on the thresholded values.

1. A squared window of interest (WOI) centered in m_k of width $2 \times (1.5r_k)$ is opened (Figure 3(b)) in $\mathbf{TV}(\mathbf{D})$. If a particle is near to one edge of the frame, the window is reduced until it falls entirely in to the frame. This reduction is not performed evenly on the two dimension: it could lead to a rectangular WOI.
2. The WOI is thresholded via a value obtained again with the Otsu method: this thresholding yields the positions of the largest changes in intensity, which are ideally located on the profile edge, and at the same time discards the fluctuations given by the residual noise (Figure 3(c)).
3. The position of the q pixels above the threshold are stored in an array $\{x_i, y_i, w_i\}_{i=1, \dots, q}$ together with the corresponding intensity values w_i .
4. A constrained regularized Least Square fit is performed (Figure 3(d)):

$$\begin{pmatrix} \tilde{\alpha}_1 \\ \tilde{\alpha}_2 \\ \tilde{\alpha}_3 \end{pmatrix} = \boldsymbol{\alpha} \sim \underset{\alpha_1^2 + \alpha_2^2 - \alpha_3 - a^2 \leq 0}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{WR}\boldsymbol{\alpha} - \mathbf{W}\mathbf{y}\|_2^2 + \frac{\mu}{2} \|\boldsymbol{\alpha}\|_2^2 \quad (2)$$

where

$$\mathbf{W} = \begin{pmatrix} \sqrt{w_1} & 0 & \dots & 0 \\ 0 & \sqrt{w_2} & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & \sqrt{w_q} \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ -2x_3 & -2y_3 & 1 \\ \vdots & \vdots & \vdots \\ -2x_q & -2y_q & 1 \end{pmatrix}, \quad \mathbf{y} = - \begin{pmatrix} x_1^2 + y_1^2 \\ x_2^2 + y_2^2 \\ \vdots \\ x_q^2 + y_q^2 \end{pmatrix}$$

and a is the true radius of the particles. The coordinates of the estimated center (x_k^e, y_k^e) are simply $(\tilde{\alpha}_1, \tilde{\alpha}_2)$, while the estimated radius r_k^e is computed as $r_k^e = \sqrt{\tilde{\alpha}_1^2 + \tilde{\alpha}_2^2 - \tilde{\alpha}_3}$: this is the main reason for the constrain in Equation (2).

The regularization term is included due to the fact that the matrix \mathbf{WR}

could be ill-conditioned [40], hence the algorithm could fail to converge to a feasible solution (e.g., if the estimated radius is greater than a): in order to avoid that, the parameter μ is set as $1/\mathcal{K}$, being \mathcal{K} the condition number of **WR**. Numerical experiments have shown that \mathcal{K} is usually large, hence μ is small, resulting on a small influence on the regularization, but still sufficient to avoid infeasible solutions. Sometimes \mathcal{K} is so large that even the regularization does not allow to achieve a feasible estimation. In this case, the regularization parameter is repeatadely increased by a factor 1.1 until the constraint is satisfied.

Remark 1. *One may wonder if a simpler procedure could be used in place of this Total Variation approach. We compared the results (on synthetic tests) obtained via our proposed approach with the ones achieved with a more direct strategy. This simple procedure estimates the center of each particle profile via the weighted mean of the elements of the connected component, while the radius is computed employing the variances of these elements. In this way, the achieved total error T is around 0.15, the Vertical error V is close to 0.10–0.11 and the Plane error P ranges between 0.08 and 0.09. Comparing these results with the one obtained via the Total Variation approach convincingly shows that the latter strategy is more effective.*

We now focus on a pathological case, where two particles are very close (Figure 4(a)): the situation is problematic, but still tractable. When the WOI is opened around one particle, it may happen that some pixels belonging to the edge of the other fall inside the window (Figure 4(a) and Figure 4(b)), affecting the least square procedure as it is evident in Figure 4(c). Thus, a further control is needed in this case. Another search for connected components is performed inside the WOI: if the number of the found components is greater than 1 (Figure 4(d)), then only the largest one is kept (Figure 4(e)). Adopting this procedure leads to a better fit, as shown in Figure 4(f).

Unfortunately, the case in Figure 5(a) can occur: the above procedure fails to recognize two distinct particles and compute a center which is very close to

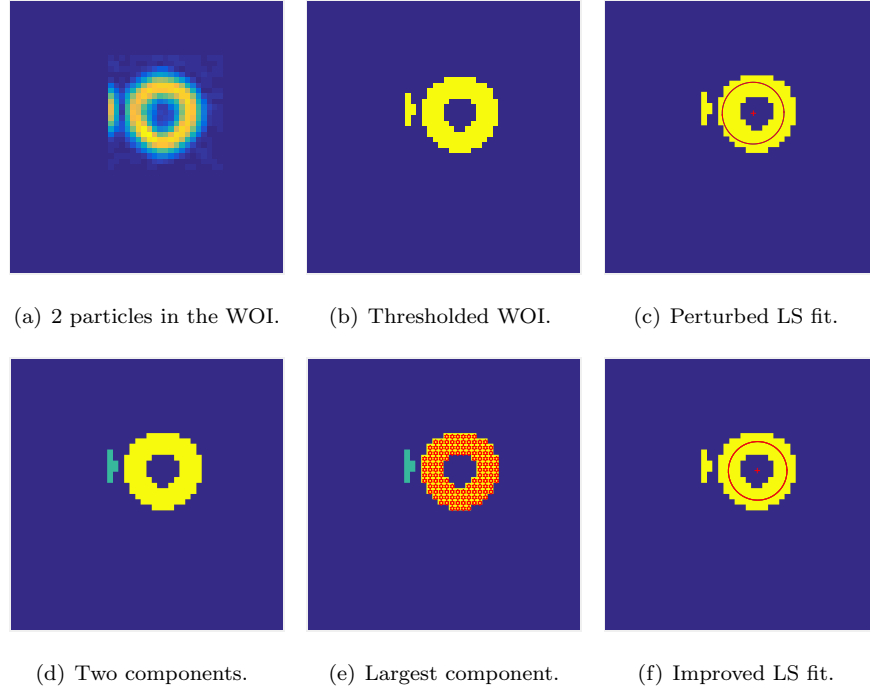


Figure 4: Upper panels: when two (or more) particles are very close but still separated, selecting a large WOI may lead to include some undesired pixels in the LS fit, resulting into a perturbed result. Bottom panels: searching inside the WOI for all connected components avoids the problem depicted in upper panels. If the particles are close but disconnected, one can easily isolate the largest component which is related to the particle, and hence a reliable LS fit can be reached.

185 the center of mass of the particles. Two possible strategy are proposed, but they still need to be investigated.

The first is to perform some morphological operations [41, 34], in order to be allowed to recognize the different particles.

The second consists of performing a LS fit using an *ellipse model*, instead of
 190 a circumference (Figure 5(c)): if the ratio of the semi-axes of the ellipse is either highly greater or lower than 1, it means that inside the ellipse there are more than one particle, due to the assumption of the spherical properties of the particles. Another check is given by the eccentricity of the ellipse. Thus, using

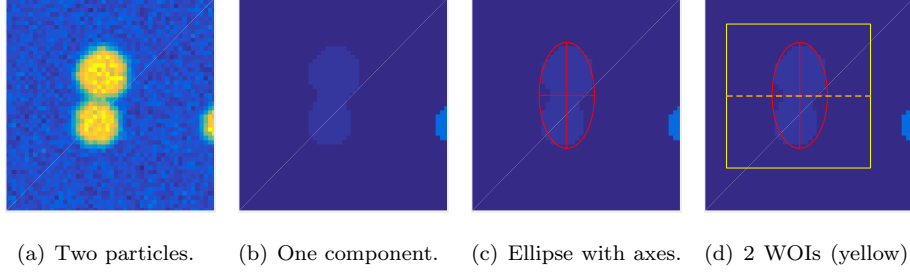


Figure 5: From left to right: true image, labelled component, estimated ellipse, WOI divided in two more WOIs. In the fourth panel, the window of interest is divided along the longest axis. The example shown refers to a vertical ellipse, but the procedure can take into account arbitrarily oriented ellipses.

the information (length and orientation) of the axes of the ellipse, the WOI can
 195 be divided in two smaller WOIs (Figure 5(d)): another LS ellipse fit is pursued
 in each portion. For each one, the ratio of the semi-axis is checked again: if it
 is around 1, then a particle is found, on the other case the same procedure is
 iterated.

Remark 2. *The situation depicted in Figure 4 can be worse: 3 or more particles
 200 can cluster, leading to an ellipsoid fit which strongly resembles a circumference.
 In this undesired case, the control on the ratio of the semi-axis could be mis-
 leading while the eccentricity can give a more reliable output. Another strategy
 could be to rely on more advanced image segmentation than simple thresholding,
 e.g. via a Mumford-Shah functional [42, 43, 44, 45, 46, 47]*

205 3.2. 3-dimensional Estimation

The procedure lying beyond lines 8–9 of Algorithm 1 for the estimation of
 the center of the particles is now explicated. It consists of two main steps: first,
 given the 2D estimation of the center of a particle in a frame, two possible 3D
 candidates are computed via the Pythagorean theorem. In a second step, we
 210 cluster all candidates belonging to the same particle.

Creation of the candidates. This procedure relies on the assumption that the radius a of the particles is known. Focussing on a single particle, assuming we have estimated its center (x^e, y^e) and the radius r^e of its circular profile in the z -th frame. The distance d between the true center and the considered frame is easily computed by $d = \sqrt{a^2 - (r^e)^2}$ (cfr. Figure 6(a)). Hence, the two candidates for the third coordinate are $zdz - d$ and $zdz + d$ (with dz the vertical discretization, equal to the separation between acquisition planes). At this point, no *prior* information is known about where the true center is located. A single particle can be spanned by Z frames, namely: hence in the ideal case Z

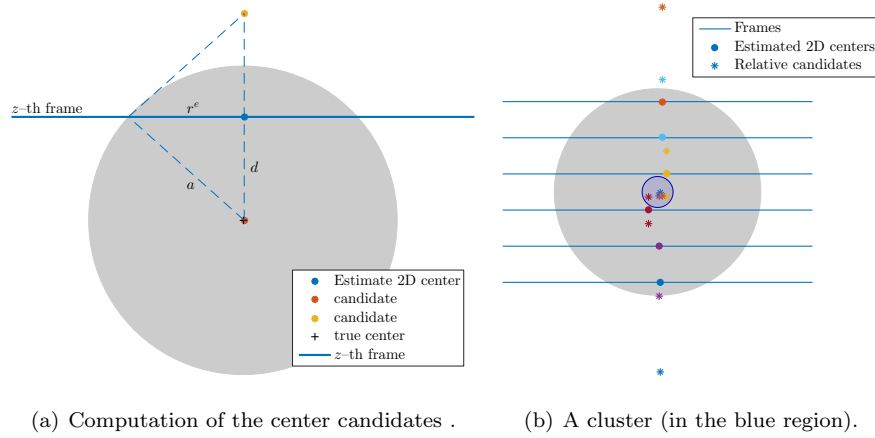


Figure 6: Panel (a): a vertical section of a particle. The horizontal line represent the z -th frame, on which an estimated center (x^e, y^e) (blue point) and estimated radius (r^e) are computed. The information on the true radius a allows to compute the distance d of the true center (black +) from the z -th frame, leading to two different candidates (red and yellow points). Panel (b): the procedure is repeated for each estimated center: in this case there are 7 frames intersecting the particle, hence 14 candidates are created. The correct ones cluster around the true center, in the highlighted circular region.

estimation for the 2D centers are available, one for each frame intersecting the particle, leading thus to have $2Z$ candidates for the true center (Figure 6(b)). Due to the geometric properties, Z candidates will cluster in a region around the true center (blue enlighten region in Figure 6(b)): the next step consists in

finding this cluster.

225 *Finding the clusters and compute the center.* For each center in each frame two candidates are created: once all the frames are processed, the situation in Figure 7 occurs. For the sake of clarity, we call \mathcal{R} the set of centers found in the frames and call \mathcal{C} the set of possible candidates computed as described in the previous paragraph (namely, the points in Figure 7(a)). It is expected that there should be a clustering around the true centers of the particles. One strategy

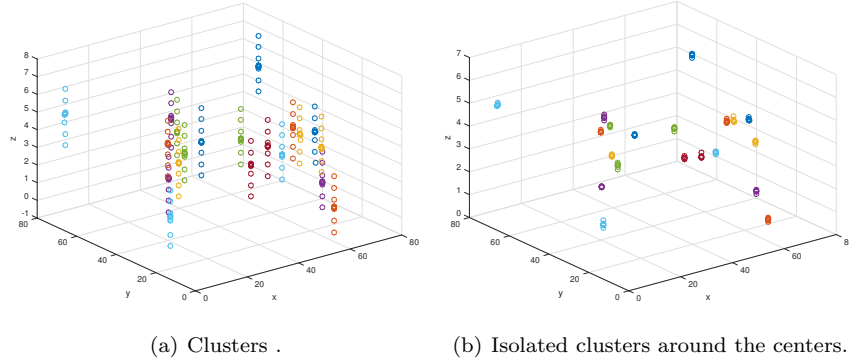


Figure 7: Left: after processing of all the frames of the volume, the clustering of the candidates around the true centers becomes evident. Right: the Z candidates which have to be used for the estimation of the center. On both figures the colors are displayed only for the sake of clarity.

230

could consist of searching for the Z nearest neighbours [48, 34] lying in a ball of radius $\rho_{\text{raw}}a$, $0 < \rho_{\text{raw}} \ll 1$ (recall that Z is the maximum number of frames spanned by a particle), but a different approach is adopted here:

1. a first raw estimation of the center of the particles is computed, using the set \mathcal{R} ;
- 235 2. the Z nearest neighbours to these approximated center are found within the candidates in \mathcal{C} .

The first step groups the points in \mathcal{R} that belong to the same particle. Once these clusters are detected and labelled, the corresponding profiles are considered and

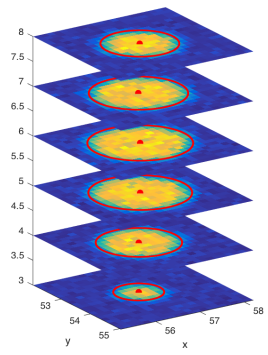
used in a LS sphere fit, in order to get a first raw estimation of the center of the particles (see Figure 8(a) for a visual inspection of this procedure). Let $\{R_i\}_{i=1,\dots,q}$ be the set of these raw estimations; focus on one of these, namely the k -th one. The Z nearest neighbours to R_k are searched within a range $\rho_{\text{est}}a$, $0 < \rho_{\text{est}} \ll 1$: let $\{(x_{ki}^e, z_{ki}^e)\}_{i=1,\dots,Z}$ be these neighbours (ideally, these are the points lying in the small highlighted circle of Figure 6(a)). The estimation of the k -th center $\mathbf{x}_k^e = (x_k^e, y_k^e, z_k^e)$ is computed as

$$x_k^e = \frac{1}{A} \sum_{i=1}^Z a_i x_{ki}^e, \quad y_k^e = \frac{1}{A} \sum_{i=1}^Z a_i y_{ki}^e, \quad z_k^e = \frac{1}{A} \sum_{i=1}^Z a_i z_{ki}^e, \quad (3)$$

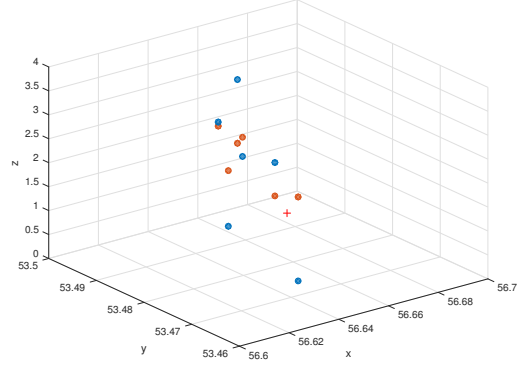
where a_i is the area of the connected component related to the center (x_{ki}^e, y_{ki}^e) (see Subsection 3.1) and $A = \sum_{i=1}^Z a_i$. A weighted mean is employed in order to
240 lower the influence on the final estimation of unreliable 2D estimations: e.g. the ones coming from frames which intersects a particle near its top or its bottom, leading to high uncertainty.

Remark 3. *It could happen that the nearest neighbours to R_k are less than Z : this can be due to low quality images, because the procedure fails to recover the*
245 *2D center in some frames or because the particle has moved during acquisition.*

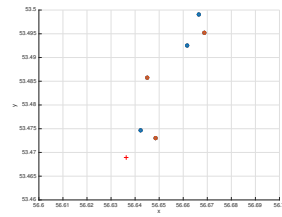
Remark 4. *The perceptive reader may wonder why the 3D procedure does not accept the LS sphere fit as final estimation of the center. Numerical experiments show that taking the LS center as final estimation leads to a total error T of $\sim 20\%$ of a voxel, which is not sufficiently precise in any real-life application,*
250 *while adopting our proposed procedure yields significantly better results. See Section 4 for the details about error measurements, performance and results.*



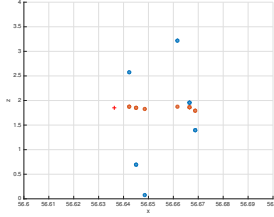
(a) Profiles for LS fit.



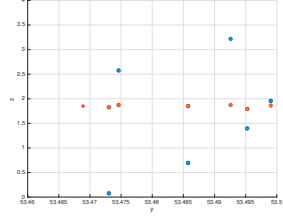
(b) Selected candidates in \mathcal{C} (orange dots).



(c) xy view.



(d) xz view.



(e) yz view.

Figure 8: (up left) overlay of the estimated center and of the circle profile of a particle over the spanned frame. The highlighted profiles are used in a LS fit to get a raw estimation of the center of the particle, indicated with the red plus in Figure 8(b). Up right: the red plus is the raw estimation of the center, the dots are the possible candidates in \mathcal{C} , the orange one are the Z nearest neighbours to the raw estimation within a range of 0.1: these points are employed in Equation (3). The reader should pay attention to the different scale of the axis. Bottom: xy , xz and yz view of the estimated center, of the candidates and of the selected candidates.

4. Numerical Tests

Two different experiments are carried on to validate the performance of the proposed algorithm. The first is devoted to evaluating the performance on synthetic datasets. Dataset construction is described in Section 2, with two different noise realization (Gaussian plus Poisson noise and pure Poisson). The evaluation is done by using three different *error measurements*, described in the subsequent paragraph. A large number of simulation are carried out,

aiming to produce a sufficient amount of data to draw reliable conclusions.
 260 Moreover, the performance of the algorithm is also evaluated on the vertical
 resolution, since this is an important issue in real-life application. The second
 experiment concerns real 3D data: it consists of considering a scanned volume
 of particles with a diameter of $3\mu\text{m}$ suspended in a glycerol/water mixture.
 Both experiments are carried on a MacBookPro, equipped with 16GB RAM
 265 and an Intel[®] Core[™] i7 CPU (2.2GHz), on MatLab 2015a. The MatLab code
 is available at <http://www-syscom.univ-mlv.fr/~benfenat/Software.html>.

Error Measurements. In order to evaluate the performance of our algorithm,
 inspired by [20, 49], three different error measurements are adopted. Denote
 with $\mathbf{c} = (c_x, c_y, c_z)^\top$ the true coordinates of a center and with $\mathbf{e} = (e_x, e_y, e_z)^\top$
 the coordinate of the relative estimation.

The *total error* T as

$$T = \sqrt{(\mathbf{c} - \mathbf{e})^\top \mathbf{D}^{-2} (\mathbf{c} - \mathbf{e})}, \quad \mathbf{D} = \begin{pmatrix} dx & 0 & 0 \\ 0 & dy & 0 \\ 0 & 0 & dz \end{pmatrix} \quad (4)$$

which aims to measure the error w.r.t. voxel precisions.

The *in-plane error* P and the *out-of-plane error* V are defined as

$$P = \sqrt{\left(\frac{c_x - e_x}{dx}\right)^2 + \left(\frac{c_y - e_y}{dy}\right)^2}, \quad V = \frac{|c_z - e_z|}{dz}. \quad (5)$$

The former aims to measure the error on the estimation of the particles' position
 in the single frames w.r.t. pixel precision, while the latter focuses on the vertical
 displacement.

270 *First synthetic test: Gaussian and Poisson noise.* Following the notation of
 Section 2, the synthetic dataset is generated using the following settings: $D_x =$
 $D_y = 76.8\mu\text{m}$, $D_z = 7\mu\text{m}$, the number N of particles is 100 of radius $a = 1\mu\text{m}$;
 the volume is discretized into a 3D array of dimension $N_x = N_y = 512$, $N_z = 22$,
 leading to voxels' dimension $dx = dy = 0.15$, $dz = 0.3182$. Two types of noise

H

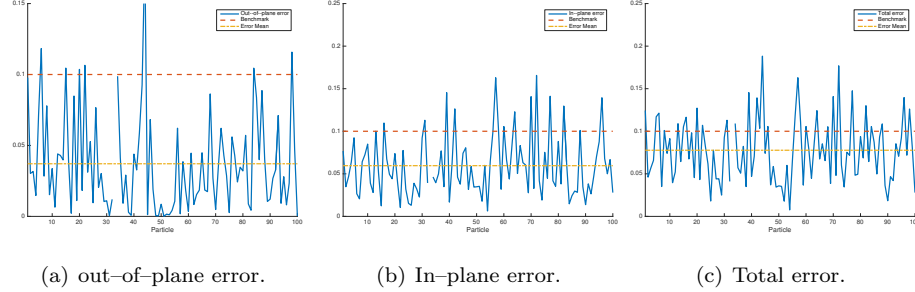


Figure 9: From left to right: V, P and T errors. Each performance stays below the state-of-art baseline, which is 10% of a pixel/voxel. The medians of the errors are 0.0289, 0.0483, and 0.0712 for V, P and T, respectively.

are affecting the frames: Gaussian ($\sigma_n = 0.2$) and Poisson (see Section 2 for the details on how the Poisson noise is added).

Algorithm 1 is applied: the chosen denoising technique (Line 2) consists simply of filtering via a Gaussian filter of dimension 5 pixels and variance 1. The window of interest is chosen as described in Subsection 3.1. Due to the discretization of the 3D volume, the maximum number Z of frames that can be spanned by a particle is 7, hence the estimation of the centers (Subsection 3.2) is achieved by

1. clustering the points in \mathcal{R} within a distance equal to $0.2a$ followed by estimating the raw center $\{R_k\}_{k=1,\dots,q}$ and then
2. search the Z nearest neighbours to each R_k within a distance $0.2a$ and apply (3).

In Figure 9 the three type of errors are depicted; the proposed procedure recognizes 99 particles (out of 100). The plots in Figure 9 show that the mean of each error (yellow dashed line) type stays *below* the 1/10 of a pixel/voxel (red line), which is the baseline of the state-of-the-art methods [20, 18]. In fact, the in-plane error is 0.0596, the out-of-plane error is 0.0371. The total error, given by (4), is 0.0777, below the state-of-the-art baseline.

In order to study the behaviour of the procedure on large numbers of particles, the above simulation is repeated 20 times (for a total of 2000 particles), storing the errors V, P, T for each run. The histograms of the total error T is shown in Figure 10(a), together with its distribution estimation. The histogram is fit with a Γ distribution with parameters (k, θ) , where k is the shape parameter and θ is the scale parameter. The mean of T is 0.0811. The behaviour of the total error is presented alone: the histogram of the in-plane error has the same appearance, with mean 0.0643, while the histogram of the out-of-plane error has also a Γ behaviour but much more concentrate towards zero, with a mean of 0.0387. All the three errors stay below the expected baseline of 10% [20]. Our proposed procedure is based on the assumption that the true radius

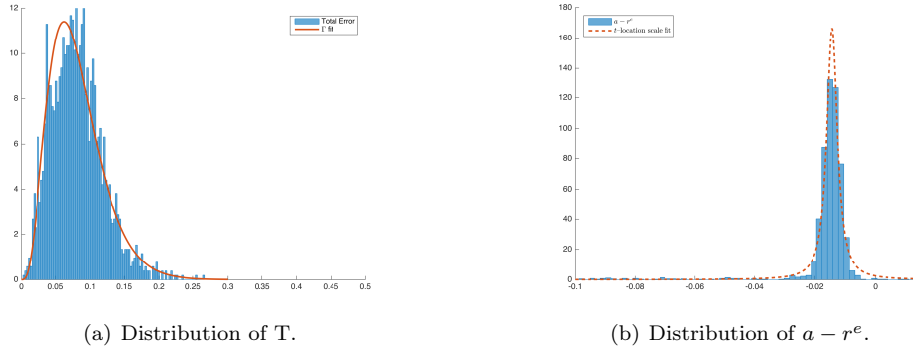


Figure 10: (a): Histogram of the total error T: its mean is 0.0811, its median is 0.0781. The out-of-plane and the in-plane error has very similar behaviour and can be fitted to the same distribution. (b): histogram of the signed difference $a - r^e$ together with its t -location scale fit. There are more outliers on the left than on the right, and in addition to the fact that the mean is circa -0.014 this tells that the proposed procedure tends to slightly overestimate the radius of the particles.

is known: this is a valid assumption in many applications, but with a certain degree of uncertainty (e.g., the radius can be known within an error of the 10%). In order to check if the estimation r^e of the radii of the particles is reliable, in Figure 10(b) the histogram of the signed difference $a - r^e$ is shown, aiming to evaluate the performance of the algorithm (r^e is computed by simple geometric

properties). The chosen distribution for the fit is the t -location scale fit, due to
the heavy tail on the left: this distribution is able to capture also the highest
error (in absolute value). In this case, there are actually some outliers on the
left of the histogram, as it is evident from Figure 10(b). The mean given by
this distribution is -0.0142: this means that overall the radii of the particles are
overestimated by 1.5%. A first justification of this behaviour can be given by
the blur effect given by the PSF (see Section 2 for the detail) combined with the
denoising technique adopted, but the next experiment will neglect the influence
of the PSF and it will show how the denoising technique influences the radius
estimation.

	N_z : number of frames						
	10	12	15	20	22	25	30
P	0.0813	0.0774	0.0719	0.0713	0.0643	0.0630	0.0620
V	0.0259	0.0301	0.0318	0.0336	0.0387	0.0471	0.0436
T	0.0883	0.0870	0.0836	0.0844	0.0811	0.0855	0.0824
$a - r^e$	-0.0117	-0.0129	-0.0141	-0.0138	-0.0142	-0.0133	-0.0137
N_{rec}	69.4	92.8	96.4	98.2	99.2	99.7	99.8

Table 1: Performance w.r.t. different vertical discretization. There is a faint decreasing
behaviour in the vertical error, which leads in a decrease on the total error. Notice that even
for a low number of frames a low V is achieved. In the last row of the table the error on
the true radius is shown for each resolution. Despite the low resolution, even for $N_z = 10$ or
 $N_z = 12$ a good estimation is achieved. The means of the differences $a - r^e$ are obtained via
a t -location scale distributio fit.

The last part is devoted to study the performance w.r.t. the vertical res-
olution, i.e. the number N_z of frames in which the volume is discretized (N_x
and N_y are unchanged, since most modern microscopes have a high resolution
in both x and y axis). In Table 1 the behaviour of the three kinds of error
are depicted for increasing vertical resolution. For each dimension, 20 different

simulations were performed, hence 20 different runs of the procedure has been
 325 done: the numbers appearing in Table 1 are the means of the results of these
 simulations. One would expect that the estimation would improve with the
 number of frames: actually, the procedure reveals itself to be very robust w.r.t.
 the vertical resolution, even with only a few (10 or 12) frames. The difference
 $a - r^e$ is depicted in the 4-th row: for each resolution, this difference is around
 330 -0.013, meaning that, regardless the number of vertical frames, the radius of
 the particles is overestimated by 1.3%. The last line of Table 1 refers to the
 (mean) number of estimated particles: the results are very satisfying for all the
 resolution but the first one ($N_z = 10$): this is due to the fact that in this case
 a particle can span only 3 frames maximum (more likely just 2 frames), leading
 335 to have a low number of candidates in \mathcal{C} . Hence, it is a problem linked to the
 relation between the dimension of the particles and the vertical resolution: for
 small particles it is sufficient to slightly increase N_z ($N_z = 12$ in order to get
 very good results), while for larger particles ($a = 1.1\mu\text{m}$) 10 frames prove to be
 sufficient, as it is evident in Table 2)

P	V	T	$a - r^e$	$N_{\text{rec}}(\%)$
0.0624	0.0262	0.0712	-0.0099	47.3 (95.5%)

Table 2: Results of 20 runs of the procedure with $a = 1.1\mu\text{m}$, $N_z = 10$ and $N = 50$. It is evident that the poor performance of the procedure when $N_z = 10$ in Table 1 is due to the relation between the diameter of the particles and the resolution. Such a low resolution is however enough for slightly larger particles to get reliable results.

340 *Second synthetic test: Poisson noise.* These tests aim at checking whether the
 Gaussian filtering is the right choice for denoising. Let consider the same setting
 of the previous experiments: $D_x = D_y = 76.8\mu\text{m}$, $D_z = 7\mu\text{m}$, 100 particles of
 radius $a = 1\mu\text{m}$, $N_x = N_y = 512$, $N_z = 22$. The difference lies in the noise
 corrupting the frames: no Gaussian noise is present ($\sigma_n = 0$) while Poisson noise
 345 affects the data. Algorithm 1 is applied to this dataset: satisfactory results,

in line with the ones in Table 1, are obtained ($P = 0.0621$, $V = 0.0331$, and $T = 0.0755$, 98 particles recognized). Since simple Gaussian filter is not always sufficient to deal with high level Poisson noise, as suggested in Subsection 3.1 an optimization approach is adopted, by using the algorithm presented in [36]:
 350 on the one hand, this procedure can be used to set the variational formulation for restoring images corrupted by pure Poisson noise and on the other to select *edge-preserving* regularization, aiming to preserve sharp edges, which eases the entire procedure of particles estimation. The Bregman procedure of [36] has been chosen instead of possibly simpler procedures (e.g., [50, 51]) for its ability
 355 to increase contrast [52, 53, 54] in the restored images, which is a desirable feature. A visual inspection on the difference between the Gaussian filtering and the employed Bregman technique is depicted in Figure 11, where a zoom of the 4-th frame is shown. The Bregman procedure uses as inner solver the AEM algorithm [55], with a maximum of 1000 iterations maximum and stopped
 360 via the criterion described in [36] with a tolerance of 10^{-4} , the fixed number of external iterations is 3, the regularization parameter μ is set to 0.1. The fit-to-data function f_0 is the generalized Kullback-Leibler and the regularization functional is the Total Variation, which preserves sharps edges.

Using this approach in Line 2 of Algorithm 1, yields the following results:
 365 $P = 0.0627$, $V = 0.0316$ and $T = 0.0752$, with 99 particles recognized. The most important difference lies in the estimated radius: with Gaussian filtering the mean error (obtained by a t -location scale fit) is -0.0134 , while the Bregman technique leads to an error of -0.0018 : hence, using the Gaussian filtering leads to overestimate the radius of the particles. Since just one single experiment is
 370 not sufficient to support this claim, further tests are carried on and presented in Table 3: one with a lower vertical resolution ($N_z = 10$), where the dimension and the discretization of the volume is the same, while the number of particles is 50 and the radius is set to $1.1\mu\text{m}$. The second test is performed on a dataset with the same characteristic of the first one presented in this paragraph: $D_x = D_y =$
 375 $76.8\mu\text{m}$, $D_z = 7\mu\text{m}$, 100 particles of radius $a = 1\mu\text{m}$, $N_x = N_y = 512$, $N_z = 22$.

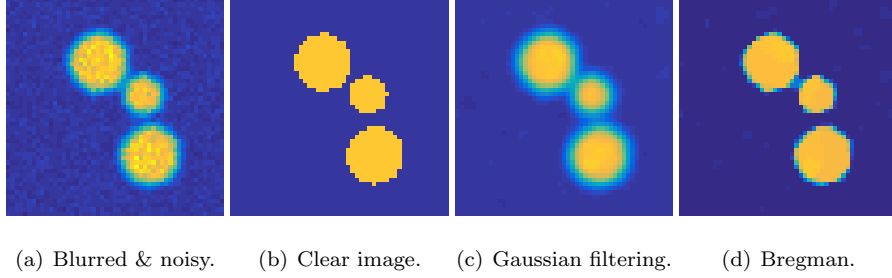


Figure 11: From left to right: Blurred & noisy frame, original image (without blurring and noise), Gaussian filtering and Bregman restoration. These images are examples from the 4-th frame, they are displayed in the range $[0, 255]$. The Bregman technique is able to separate in a more reliable way the particles and at the same time is providing with more sharp edges, due to the choice of regularization function. In this case, the regularization is given by the Total Variation functional. It could happen that Gaussian filtering makes merge two or more particle in one big component, increasing the difficulties in recognized different objects.

Table 3 shows that using the correct denoising procedure produces better results in terms of error estimation and of number of recognized particles; moreover, choosing the correct denoising technique allows to estimate more precisely the radius: in fact, for $N_z = 10$ using Gaussian filtering leads to an error of almost 1%, while the Bregman technique reduces the error to 0.1%. For $N_z = 22$ the difference is more pronounced: classical filtering gives an error of $\sim 1.4\%$, while again the proposed approach results in an error of only 0.1%. The hypothesis that the overestimation of the radius actually depends on the denoising and deblurring technique is true: at a first sight, it seems from Table 1 that this is a *determinate error* [20] of the algorithm, but this last experiment tells the opposite. The procedure used to improve the quality of the images influences the performance of the particle estimation algorithm.

While on the one hand, the two denoising procedures are similar, because both require parameters setting (e.g., the Bregman technique requires the tuning of the regularization parameter, of the tolerance for the stopping criterion; the filtering techniques requires to choose the type of filter and its parameters); on

N_z	Technique	P	V	T	$a - r^e$	$N_{\text{rec}}(\%)$
10	Gaussian Filter	0.0622	0.0271	0.0712	-0.0096	46.7 (93.4%)
	Bregman	0.0628	0.0177	0.0677	-0.0013	47.6 (95.2%)
22	Gaussian Filter	0.0626	0.0380	0.0793	-0.0138	98.8 (98.8%)
	Bregman	0.0637	0.0291	0.0746	-0.0019	99.4 (99.4%)

Table 3: Results obtained by 10 runs of the algorithm. The Bregman technique provides better results overall, both for $N_z = 10$ and $N_z = 22$. The error on the estimated radius is given by the mean obtained by the t -location scale distribution fit, as was done in Figure 10(b); for the case $N_z = 10$, looking at the simple arithmetic mean, the Bregman procedure shows to be much more precise in the radius' estimation, in fact it gives an error of -0.0028 , while the Gaussian filtering results in an error of -0.0119 . For the case $N_z = 22$. The overall behaviour of the Bregman approach in terms of error measurements is slightly better, but the number of found particles is closer to the maximum and the estimation of the radius improved, reaching an error of 0.1%.

the other hand, the optimization technique has drawbacks as its computational cost and the time need to restore each frame, while simple filtering is more or less free in these terms. There is a trade-off (as it usually occurs in cases such these) between performance and time/computational cost.

Real 3D data. This paragraph is devoted to applying the proposed algorithm to real 3D data. The scanned volume has $D_x = D_y = 64\mu\text{m}$, $D_z = 4.1\mu\text{m}$, discretized into an array of dimension $512 \times 512 \times 10$, leading to $dx = dy = 0.125\mu\text{m}$, $dz = 0.41\mu\text{m}$; 50 scans of the volume were recorded, with a $dt = 0.5\text{s}$. The diameter of the particles is $3\mu\text{m}$ ($a = 1.5\mu\text{m}$) and they are suspended in a $\sim 70\% - 30\%$ glycerol/water mixture (viscosity of $\sim 0.017 \text{ Pa s}$). The instrument used to acquire this data is a confocal microscope (Zeiss LSM 700) with a $100\times\text{NA } 1.4$ oil immersion objective (Zeiss Plan-APOCHROMAT). The frames are restored using the Bregman procedure previously described with the following settings: AEM as inner solver with a Total Variation functional as regularization, maximum number of allowed iterations set to 1000 within a tol-

erance of 10^{-4} for the stopping criterion described in [36] with $\alpha = 2$, 3 external
iteration are allowed. Since the images are given without any information about
410 their recording, a Gaussian PSF¹ with $\sigma = 1$ and zero mean is assumed as blur-
ring operator, a background term equal to the minimum value of the image,
and Poisson noise affecting the frames. All these assumptions are consistent
with the type of the images produced by the aforementioned instrument. Fig-

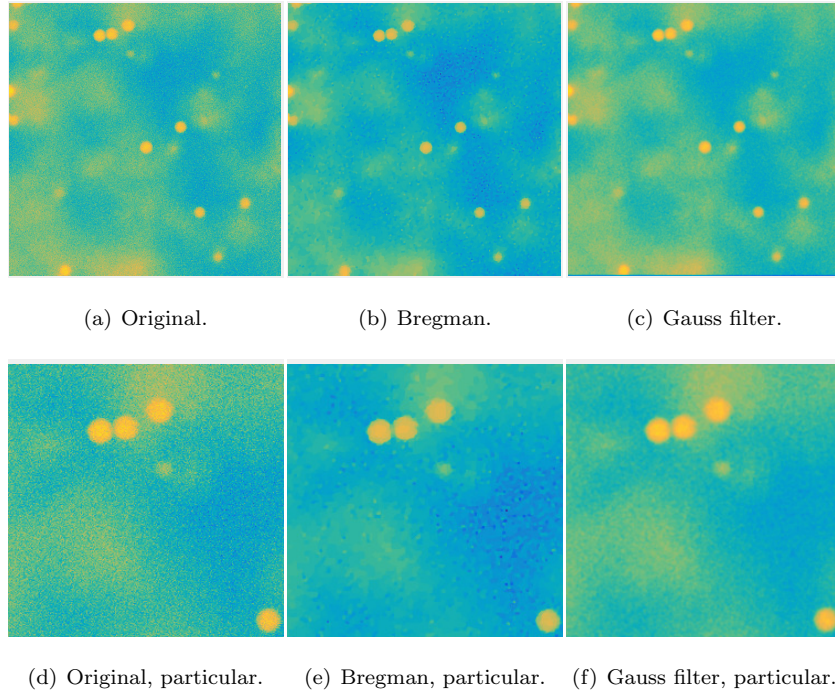


Figure 12: From left to right, from up to bottom. The images have to be read in pairs: for the z -th slice, the left image refers to the noisy and blurred frame, while the right image refers to the restored one. It is clear that the contrast of the image is significantly improved, reducing the diffuse areas, mainly in the highest frames. All the images are displayed in the range $[0, 255]$.

ure 12 shows in its first row the 6-th acquired frame at time $t = 1$, the restored

¹`fspecial('gaussian',512,1).`

415 version via Bregman technique and the filtered image via a Gaussian filter. In
the second row a particular of these image is presented: the visual inspection
makes clear that the usage of the correct denoising technique allows to reduce
the glowing halo all among the frame and moreover provides with more sharp
edges, all this contributes in making easier the recognition of the profiles.

420 Algorithm 1 is set with an initial woi of width $2 \times 0.1r_k$ (see Subsection 3.1
for the details), with a threshold which is 1.5 times the value given by Otsu's
method, $\rho_{\text{raw}} = 0.3$, $\rho_{\text{est}} = 0.3$. The frames at time $t = 1$ are shown in Figure 14:
the goodness of result is more evident in the particular shown in the last panel.

Figure 13 provides a visual inspection of the reconstructed position of the
425 particles at time $t = 1$: this reconstruction faithfully respects the true position,
as it is clear by comparing the 3D plot with the frames depicted in Figure 14,
where the recovered profiles of the particles are superimposed on the original
images. In these images, the top left corner corresponds to the point $(0, 0, kdz)$
in the 3D space, being k the number of the frame. A closer inspection of
430 Figure 13 and 14 demonstrates that the proposed procedure finds particles close
to the boundaries of the frames, as well as the ones near the top or the bottom
of the volume.

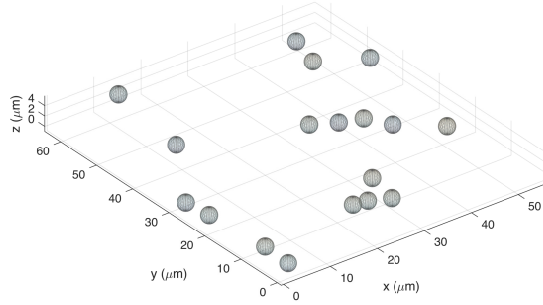
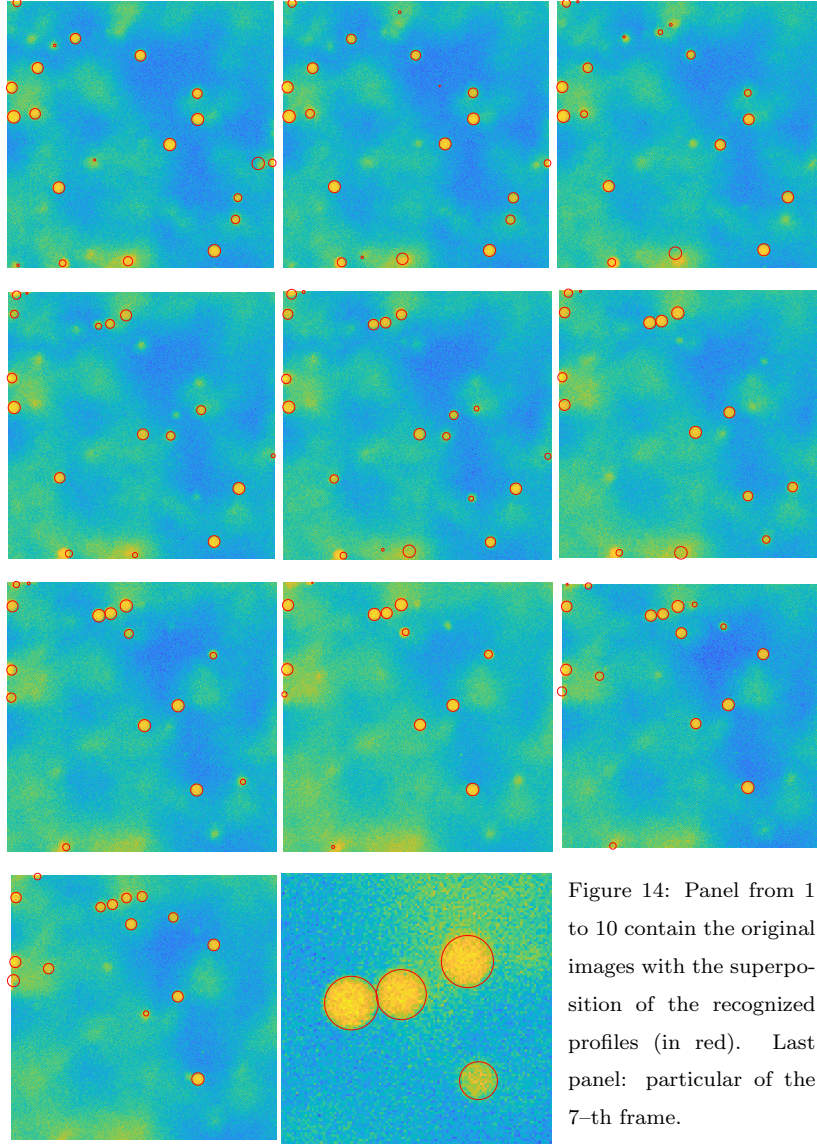


Figure 13: 3D recovering of the position of the particles at time $t = 1$.



5. Conclusion

In this work, a particle segmentation and position estimation methodology is presented. Assuming fixed spherical particles with a known radius, this procedure uses the 2D gradient information on the profiles of the particles and

employs a weighted regularized Least Square fit to find the 2D center and the
 radius of the profile intersecting each frame. Using geometric properties, the
 coordinates of the 3D center are retrieved with an accuracy better than 10% of
 440 a voxel, which was the state-of-the-art performance of this type of algorithms.
 Even for a low vertical resolution, the total error is still acceptable, and the very
 low error on the radius estimation suggests that this procedure can improve *a*
priori information about the radius of particles of uncertain dimension. The
 presence of certain type of noise requires a more tailored denoising technique: in
 445 this work it is suggest to employ image restoration technique, which can be more
 demanding in terms of computational time but this strategy leads to a general
 improvement of the position estimation. Moreover, this approach significantly
 increases the precision on the radius estimation. Future work will involve better
 thresholding techniques for pathological cases, handling the case of the unknown
 450 radius; the case where particles are subjected to significant Brownian motion,
 and the proposed algorithm will be employed in a more general particle tracking
 procedure.

References

- [1] J. Gelles, B. J. Schnapp, M. P. Sheetz, Tracking kinesin-driven movements
 455 with nanometre-scale precision, *Nature* 331 (1988) 450 EP – (1988).
- [2] A. Yildiz, J. N. Forkey, S. A. McKinney, T. Ha, Y. E. Goldman, P. R.
 Selvin, Myosin V walks hand-over-hand: single fluorophore imaging with
 1.5-nm localization, *Science* 300 (5628) (2003) 2061–2065 (Jun 2003).
- [3] B. Brandenburg, X. Zhuang, Virus trafficking-learning from single-virus
 460 tracking, *Nature reviews. Microbiology* 5 (3) (2007) 197–208 (03 2007).
 doi:10.1038/nrmicro1615.
- [4] G. Seisenberger, M. U. Ried, T. Endreß, H. Büning, M. Hallek, C. Bräuchle,
 Real-time single-molecule imaging of the infection pathway of an adeno-

- associated virus, *Science* 294 (5548) (2001) 1929–1932 (2001). doi:10.1126/science.1064103.
- [5] A. Akhmanova, M. O. Steinmetz, Tracking the ends: a dynamic protein network controls the fate of microtubule tips, *Nature Reviews Molecular Cell Biology* 9 (2008) 309 EP – (04 2008).
- [6] A. Kusumi, T. A. Tsunoyama, K. M. Hirose, R. S. Kasai, T. K. Fujiwara, Tracking single molecules at work in living cells, *Nature Chemical Biology* 10 (2014) 524 EP – (06 2014). doi:10.1038/nchembio.1558.
- [7] M. Weiss, Chapter Eleven—crowding, diffusion, and biochemical reactions, in: R. Hancock, K. W. Jeon (Eds.), *New Models of the Cell Nucleus: Crowding, Entropic Forces, Phase Separation, and Fractals*, Vol. 307 of *International Review of Cell and Molecular Biology*, Academic Press, 2014, pp. 383 – 417 (2014). doi:10.1016/B978-0-12-800046-5.00011-4.
- [8] M. J. Saxton, K. A. Jacobson, Single-particle tracking: applications to membrane dynamics., *Annual review of biophysics and biomolecular structure* 26 (1997) 373–99 (1997).
- [9] U. Jandt, A.-P. Zeng, *Modeling of Intracellular Transport and Compartmentation*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 221–249 (2012). doi:10.1007/10_2011_104.
- [10] L. L. Josephson, J. W. Swan, E. M. Furst, In situ measurement of localization error in particle tracking microrheology, *Rheologica Acta* (2018). doi:10.1007/s00397-018-1117-5.
- [11] T. G. Mason, D. A. Weitz, Optical measurements of frequency-dependent linear viscoelastic moduli of complex fluids, *Phys. Rev. Lett.* 74 (1995) 1250–1253 (1995). doi:10.1103/PhysRevLett.74.1250.
- [12] D. T. Chen, E. R. Weeks, J. C. Crocker, M. F. Islam, R. Verma, J. Gruber, A. J. Levine, T. C. Lubensky, A. G. Yodanis, *Rheological microscopy: Lo-*

cal mechanical properties from microrheology, *Phys. Rev. Lett.* 90 (2003) 108301 (Mar 2003). doi:10.1103/PhysRevLett.90.108301.

- [13] J. Apgar, Y. Tseng, E. Fedorov, M. B. Herwig, S. C. Almo, D. Wirtz,
Multiple-particle tracking measurements of heterogeneities in solutions of
actin filaments and actin bundles., *Biophysical Journal* 79 (2) (2000) 1095–
1106 (08 2000).
- [14] M. T. Valentine, P. D. Kaplan, D. Thota, J. C. Crocker, T. Gisler, R. K.
Prud’homme, M. Beck, D. A. Weitz, Investigating the microenvironments
of inhomogeneous soft materials with multiple particle tracking, *Phys. Rev.*
E 64 (2001) 061506 (Nov 2001). doi:10.1103/PhysRevE.64.061506.
- [15] K. K. Chu, D. Mojahed, C. M. Fernandez, Y. Li, L. Liu, E. J. Wilsterman,
B. Diephuis, S. E. Birket, H. Bowers, G. M. Solomon, B. S. Schuster,
J. Hanes, S. M. Rowe, G. J. Tearney, Particle-tracking microrheology using
micro-optical coherence tomography, *Biophysical Journal* 111 (5) (2016)
1053 – 1063 (2016). doi:https://doi.org/10.1016/j.bpj.2016.07.020.
- [16] C. A. Murray, W. O. Sprenger, R. A. Wenk, Comparison of melting in
three and two dimensions: Microscopy of colloidal spheres, *Phys. Rev. B*
42 (1990) 688–703 (Jul 1990). doi:10.1103/PhysRevB.42.688.
- [17] J. C. Crocker, D. G. Grier, Microscopic measurement of the pair interaction
potential of charge-stabilized colloid, *Phys. Rev. Lett.* 73 (1994) 352–355
(Jul 1994). doi:10.1103/PhysRevLett.73.352.
- [18] N. Chenouard, I. Smal, F. de Chaumont, M. Maška, I. F. Sbalzarini,
Y. Gong, J. Cardinale, C. Carthel, S. Coraluppi, M. Winter, A. R. Co-
hen, W. J. Godinez, K. Rohr, Y. Kalaidzidis, L. Liang, J. Duncan,
H. Shen, Y. Xu, K. E. G. Magnusson, J. Jaldén, H. M. Blau, P. Paul-
Gilloteaux, P. Roudot, C. Kervrann, F. Waharte, J.-Y. Tinevez, S. L.
Shorte, J. Willemsse, K. Celler, G. P. van Wezel, H.-W. Dan, Y.-S. Tsai,
C. O. de Solórzano, J.-C. Olivo-Marin, E. Meijering, Objective comparison
of particle tracking methods, *Nature Methods* 11 (2014) 281 EP – (2014).

- 520 [19] T. Savin, P. S. Doyle, Static and dynamic errors in particle tracking microrheology, *Biophysical Journal* 88 (1) (2005) 623 – 638 (2005). doi:
<https://doi.org/10.1529/biophysj.104.042457>.
- [20] M. K. Cheezum, W. F. Walker, W. H. Guilford, Quantitative comparison of algorithms for tracking single fluorescent particles., *Biophysical Journal* 81 (4) (2001) 2378–2388 (10 2001).
525
- [21] A. Kusumi, Y. Sako, M. Yamamoto, Confined lateral diffusion of membrane receptors as studied by single particle tracking (nanovid microscopy). Effects of calcium-induced differentiation in cultured epithelial cells., *Biophysical Journal* 65 (5) (1993) 2021–2040 (11 1993).
- 530 [22] W. H. Guilford, R. W. Gore, The mechanics of arteriole-tissue interaction, *Microvascular Research* 50 (2) (1995) 260 – 287 (1995). doi:10.1006/
mvre.1995.1058.
- [23] R. N. Ghosh, W. W. Webb, Automated detection and tracking of individual and clustered cell surface low density lipoprotein receptor molecules.,
535 *Biophysical Journal* 66 (5) (1994) 1301–1318 (05 1994).
- [24] G. M. Lee, A. Ishihara, K. A. Jacobson, Direct observation of Brownian motion of lipids in a membrane, *Proceedings of the National Academy of Sciences* 88 (14) (1991) 6274–6278 (1991). doi:10.1073/*pnas*.88.14.
6274.
- 540 [25] J. C. Crocker, D. G. Grier, Methods of digital video microscopy for colloidal studies, *Journal of Colloid and Interface Science* 179 (1) (1996) 298 – 310 (1996).
- [26] M. Jenkins, S. Egelhaaf, Confocal microscopy of colloidal particles: Towards reliable, optimum coordinates, *Advances in Colloid and Interface Science* 136 (1) (2008) 65 – 92 (2008). doi:10.1016/*j.cis*.2007.07.006.
545
- [27] C. Anderson, G. Georgiou, I. Morrison, G. Stevenson, R. Cherry, Tracking of cell surface receptors by fluorescence digital imaging microscopy using a

- charge-coupled device camera. Low-density lipoprotein and influenza virus receptor mobility at 4 degrees C, *Journal of Cell Science* 101 (2) (1992) 415–425 (1992).
- [28] G. J. Schütz, H. Schindler, T. Schmidt, Single-molecule microscopy on model membranes reveals anomalous diffusion., *Biophysical Journal* 73 (2) (1997) 1073–1080 (08 1997).
- [29] I. Sbalzarini, P. Koumoutsakos, Feature point tracking and trajectory analysis for video imaging in cell biology, *Journal of Structural Biology* 151 (2) (2005) 182 – 195 (2005). doi:10.1016/j.jsb.2005.06.002.
- [30] M. Husain, T. Boudier, P. Paul-Gilloteaux, I. Casuso, S. Scheuring, Software for drift compensation, particle tracking and particle analysis of high-speed atomic force microscopy image series, *Journal of Molecular Recognition* 25 (5) (2012) 292–298 (2012). doi:10.1002/jmr.2187.
- [31] W. J. Godinez, K. Rohr, Tracking multiple particles in fluorescence time-lapse microscopy images via probabilistic data association, *IEEE Transactions on Medical Imaging* 34 (2) (2015) 415–432 (Feb 2015). doi:10.1109/TMI.2014.2359541.
- [32] K. E. G. Magnusson, J. Jaldén, A batch algorithm using iterative application of the Viterbi algorithm to track cells and construct cell lineages, in: 2012 9th IEEE International Symposium on Biomedical Imaging (ISBI), 2012, pp. 382–385 (May 2012). doi:10.1109/ISBI.2012.6235564.
- [33] P. C. Hansen, J. G. Nagy, D. P. O’Learly, Deblurring images: matrices, spectra, and filtering, *SIAM*, 2006 (2006).
- [34] R. C. Gonzalez, R. E. Woods, *Digital Image Processing* (2nd Edition), Pearson, 2002 (2002).
- [35] M. Bertero, P. Boccacci, G. Desiderà, G. Vicidomini, Image deblurring with Poisson data: from cells to galaxies, *Inverse Problems* 25 (2009) 123006 (2009).

- [36] A. Benfenati, V. Ruggiero, Inexact Bregman iteration with an application to Poisson data reconstruction, *Inverse Problems* 29 (6) (2013) 065016 (2013).
- [37] R. Zanella, P. Boccacci, L. Zanni, M. Bertero, Efficient gradient projection methods for edge-preserving removal of Poisson noise, *Inverse Problems* 25 (2009) 045010 (2009).
- [38] M. A. T. Figueiredo, J. M. Bioucas-Dias, Deconvolution of Poissonian images using variable augmented Lagrangian optimization, *IEEE Workshop on Statistical Signal Processing*, Cardiff (2009).
- [39] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Transactions on Systems, Man, and Cybernetics* 9 (1) (1979) 62–66 (Jan 1979). doi:10.1109/TSMC.1979.4310076.
- [40] G. Golub, C. Van Loan, *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 2013 (2013).
- [41] É. Puybareau, H. Talbot, N. Gaber, T. Bourouina, Morphological analysis of Brownian motion for physical measurements, in: J. Angulo, S. Velasco-Forero, F. Meyer (Eds.), *Mathematical Morphology and Its Applications to Signal and Image Processing*, Springer International Publishing, Cham, 2017, pp. 486–497 (2017).
- [42] L. Ambrosio, V. M. Tortorelli, Approximation of functional depending on jumps by elliptic functional via t-convergence, *Communications on Pure and Applied Mathematics* 43 (8) (1990) 999–1036 (1990). doi:10.1002/cpa.3160430805.
- [43] D. Mumford, J. Shah, Optimal approximations by piecewise smooth functions and associated variational problems, *Communications on Pure and Applied Mathematics* 42 (5) (1989) 577–685 (1989). doi:10.1002/cpa.3160420503.
URL 10.1002/cpa.3160420503

- [44] M. Foare, J. O. Lachaud, H. Talbot, Image restoration and segmentation
 605 using the Ambrosio–Tortorelli functional and discrete calculus, in: 2016
 23rd International Conference on Pattern Recognition (ICPR), 2016, pp.
 1418–1423 (Dec 2016). doi:10.1109/ICPR.2016.7899836.
- [45] A. Vitti, The Mumford–Shah variational model for image segmentation:
 An overview of the theory, implementation and use, ISPRS Journal of
 610 Photogrammetry and Remote Sensing 69 (2012) 50 – 64 (2012). doi:
 10.1016/j.isprsjprs.2012.02.005.
- [46] C.-K. Yap, H. K. Lee, Identification of cell nucleus using a Mumford-Shah
 ellipse detector, in: Advances in Visual Computing, Springer Berlin Hei-
 delberg, Berlin, Heidelberg, 2008, pp. 582–593 (2008).
- [47] R. Zanella, F. Porta, V. Ruggiero, M. Zanetti, Serial and parallel ap-
 615 proaches for image segmentation by numerical minimization of a second-
 order functional, Appl. Math. Comput. 318 (C) (2018) 153–175 (Feb. 2018).
 doi:10.1016/j.amc.2017.07.021.
- [48] I. Kononenko, M. Kukar, Machine Learning and Data Mining: Introduction
 620 to Principles and Algorithms, Horwood Publishing Limited, 2007 (2007).
- [49] T. Savin, P. S. Doyle, Static and dynamic errors in particle tracking mi-
 crorheology, Biophysical Journal 88 (1) (2005) 623 – 638 (2005). doi:
 10.1529/biophysj.104.042457.
- [50] S. Bonettini, A. Benfenati, V. Ruggiero, Primal-dual first order methods for
 625 total variation image restoration in presence of Poisson noise, in: 2014 IEEE
 International Conference on Image Processing (ICIP), 2014, pp. 4156–4160
 (2014). doi:10.1109/ICIP.2014.7025844.
- [51] S. Bonettini, A. Benfenati, V. Ruggiero, Scaling techniques for ε – sub-
 gradient methods, SIAM Journal on Optimization 26 (3) (2016) 1741–1772
 630 (2016). doi:10.1137/14097642X.

- [52] A. Benfenati, V. Ruggiero, Image regularization for Poisson data, *Journal of Physics: Conference Series* 657 (1) (2015) 012011 (2015).
- [53] A. Benfenati, A. La Camera, M. Carbillat, Deconvolution of post-adaptive optics images of faint circumstellar environments by means of the inexact Bregman procedure, *Astronomy & Astrophysics* 586 (2016) A16 (2016).
doi:10.1051/0004-6361/201526960.
- [54] A. Benfenati, V. Ruggiero, Inexact Bregman iteration for deconvolution of superimposed extended and point sources, *Communications in Nonlinear Science and Numerical Simulation* 21 (1) (2015) 210-224 (2015).
- [55] S. Bonettini, V. Ruggiero, An alternating extragradient method for total variation-based image restoration from poisson data, *Inverse Problems* 27 (9) (2011) 095001 (2011).

Alessandro Benfenati received his Ph.D. in Mathematics from University of Ferrara in 2015. He was Post-doc at Math and CS department, University of Ferrara, (2015-2016), at LIGM, Université Paris-est (2016-2017) and at ESIEE Paris (2017-2018). He is researcher at ESP department, University of Milan. His interest include optimization, regularization techniques.

Francesco Bonacci received his physics degree at the University of Perugia (2016). He is now Ph.D. student in the Navier laboratory (UMR 8205) in Marne-La-Vallée, Paris. He started to work on the rheology of dense colloidal suspensions. His general research interests involve the physics of soft materials (colloids, emulsions, polymers).

Tarik Bourouina holds M.Sc. in Physics, M. Eng. in Optoelectronics, Ph.D. in MEMS (1991), and HDR (2000) from Université Paris-Sud, Orsay. Since 2002 Dr. Bourouina is Professor at ESIEE Paris, Université Paris-Est, appointed as Dean for Research from 2012 to 2015. His current interests include optofluidics, analytical chemistry on-chip.

Hugues Talbot received the Habilitation from Université Paris-Est (2013), the PhD from Ecole des Mines de Paris (1993) and the engineering degree from

⁶⁶⁰ Ecole Centrale de Paris (1989). He was PRS at CSIRO, Sydney (1994-2004).
He is professor at CentraleSupélec, Université Paris-Saclay. His interests include
mathematical morphology, combinatorial and continuous optimization.