Efficient position estimation of 3D fluorescent spherical beads in confocal microscopy via Poisson denoising

Alessandro Benfenati, Francesco Bonacci, Tarik Bourouina, Hugues Talbot

Journal of Mathematical Imaging and Vision manuscript No.
(will be inserted by the editor)

Efficient position estimation of 3D fluorescent spherical beads in confocal microscopy via Poisson denoising

Alessandro Benfenati · Francesco Bonacci · Tarik Bourouina · Hugues Talbot

Received: date / Accepted: date

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 imaging data: this work presents a technique to estimate the 3D coordinates of the center of spherical particles. This procedure has its core in the processing of the images of the scanned volume: it firstly applies denoising techniques to each frame of the scanned volume, then 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. Then, the 2D information is used to retrieve the 3D coordinates using geometrical properties. The experiments provide evidence that image denoising has a large impact on the performance of the particle tracking procedures, since they strongly depend on the quality of the initial acquisition. This work shows that the choice of tailored image denoising technique for Poisson noise 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.

Alessandro Benfenati
Dipartimento di Scienze e Politiche Ambientali, Via Celoria 2, 20133, Milano, Italy
E-mail: alessandro.benfenati@unimi.it

Francesco Bonacci
IFSTTAR, ENPC, Université Paris-Est, 14–20 Boulevard Newton, 77420, Champs sur Marne, France
E-mail: francesco.bonacci@ifsttar.fr

Tarik Bourouina
ESYCOM, ESIEE, ENPC, Université Paris-Est, 5 Boulevard Descartes 77420 Champs-sur-Marne, France E-mail: tarik.bourouina@esiee.fr

Hugues Talbot
CVN, Inria, CentraleSupélec, Université Paris-Saclay,9 Rue Joliot Curie, 91190, Gif-sur-Yvette, France
E-mail: hugues.talbot@centralesupelec.fr

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 [42], in the study of the infection path of a virus [39] or in the investigation of cytoskeletal filaments [1]; another topic involving particles tracking problem regards the observation of protein motion in cell membranes [29] or intracellular transport [24]. Other interesting areas of application include statistical mechanics [4,5], fluid dynamics and mechanics, in particular Rheology [26], where the thermal motion of Brownian particles has been tracked to study local rheological properties [14]; complex fluids [2,37]; and Microrehology in Medicine [17]. Colloidal works have benefited from developments in particle tracking procedures in measuring biofluids such as mucus [38] and vitreous humor [40]. All these practical instances of particle tracking rely on imaging data, acquired via confocal microscopy, electric microscopy and/or similar techniques.

It has been pointed out [15] 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. The former is based on the acquired images, while the latter employs the retrieved information together with probabilistic results. In the past, several procedure have been proposed to estimate the particle position: cross–correlation of a sequence of images [27], centroid techniques [25], Gaussian fitting [30]. Some of them claim subpixel resolution, and in [13] a wide comparison of these techniques showed that significant numerical experimentation is needed before validating such results. Other methods includes combinatorial optimization [36], nearest neighbour [23], Kalman filtering coupled with probabilistic data association [20], use of the Viterbi algorithm [31] and several others. An experimental comparison of a plethora of methods can be found in [15]. In [34] (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 [13] 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 analysis focuses also on the role of image denoising techniques, which heavily influence the final result and performance of position estimation algorithms. The proposed procedure aims mainly to treat the *static error* [34], which arise from noise affecting this type of experiments; this *static error* is equivalent to the notion of *precision* in [13].

Following [34] and the consideration in [13] about preliminary synthetic experiments, 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. 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 presented procedure 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. In real life application, 3D confocal data are corrupted by noise, usually of Poisson type, hence denoising techniques are necessary to ensure the good quality of the reconstruction. In this work the comparison between classical Gaussian filtering and more tailored algorithm for noise removal is done.

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 $\mathcal{N}\left(\mu, \sigma^2\right)$ indicates a Gaussian distribution of mean μ and variance σ^2. $\mathbf{I}$ denotes the identity matrix, $\mathbf{0}$ the vector with all zeros entries.

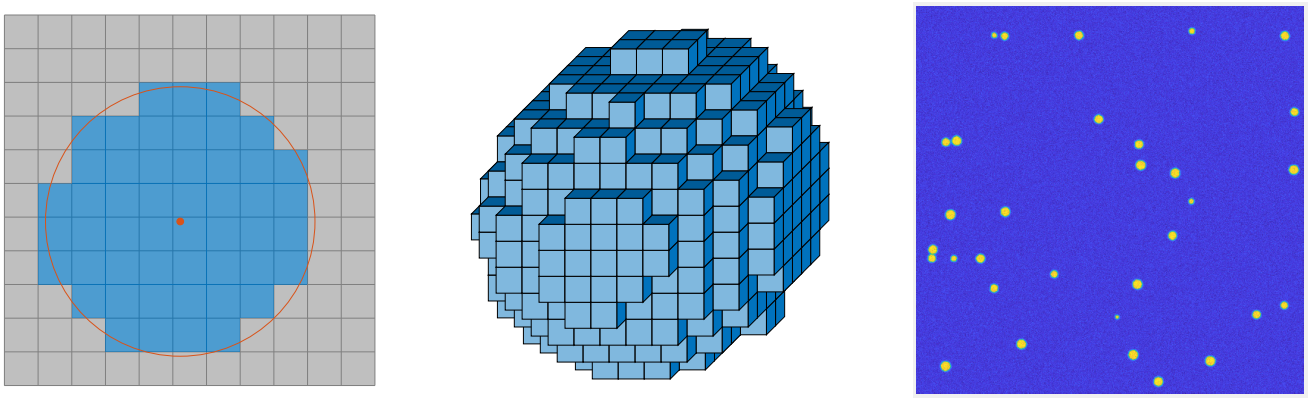
2 Data Creation: Simulation Procedure

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 dimension $D_x \times D_y \times D_z$. The particles are assumed to have all the same, known radius a;
- the 3D volume is discretized into an array of $N_x \times N_y \times N_z$ voxels; each voxel has dimension $\mathrm{d}x \times \mathrm{d}y \times \mathrm{d}z$, being $\mathrm{d}x = D_x/N_x, \mathrm{d}y = D_y/N_y, \mathrm{d}z = D_z/N_z$. N_z represents the number of 2D frames. Each particle is discretized 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.

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 $\mathbf{x}_i = (x_i, y_i, z_i)^\top$.



(a) Discretization, 2D. (b) Discretization, 3D. (c) Frame.

Fig. 2.1 Panel 2.1(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 is not always possible to discretize the disk in a symmetric fashion. The procedure follows the same ratio for the 3D case (2.1(b)). Panel 2.1(c): blurred and noisy frame

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, 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 2.1(a) a 2D explanation of this procedure is depicted: the 3D case follows the same procedure (Figure 2.1(b)).

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 the added to each frame: let $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \sigma_n \mathbf{I})$ be 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 [22])

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

being $\mathbf{F}_z$ the z–th frame and $\|\cdot\|_{\mathrm{F}}$ the Froebenius norm. A different noise realization $\boldsymbol{\eta}$ 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 `F` the current frame (see the MatLab help for the `imnoise` function for more details about this procedure.). Finally, the intensity values of each frame are rescaled into the interval $[0, 255]$. See Figure 2.1(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:

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 (lines 8–9).

3.1 Frames Processing

The procedures 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: this procedure is very quick and inexpensive, performed via the FFT MatLab's native algorithms, see Figure 3.1(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 denoising strategy based on an optimization method: given the noisy and blurred frame $\mathbf{g}$,

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

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

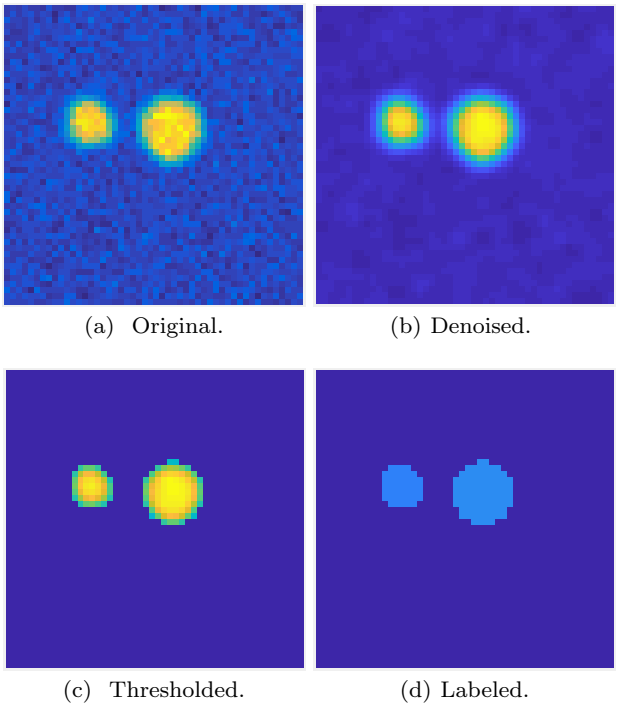
one is led to compute the denoised frame $\tilde{\mathbf{f}}$ as

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

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 Point Spread Function (PSF), e.g. the linear blurring operator, b is a constant background 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 [6,43,18]. 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 as in the current framework) 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 [32] (see Figure 3.1(c)). Then, the K connected components $\{L_k\}_{k=1,\dots,K}$ in the thresholded frame are recognized and labeled (Figure 3.1(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 Equation (3.3)). 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 3.2(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 [43] is applied to the current denoised frame, namely $\mathbf{D}$, aiming to find the edges of the particles (Figure 3.2(c)). Denoting (with an abuse of notation) the partial derivatives



(a) Original. (b) Denoised. (c) Thresholded. (d) Labeled.

Fig. 3.1 Particular of the frame of Figure 2.1(c). (a): a region of interest with two separated particles. (b): result of the Gaussian filtering. The noise is reduced, but the edges are blurred. (c): thresholding via the Otsu method. (d): labeling procedure, where different colors mean different labels. The order of labeling does not influence the final result.

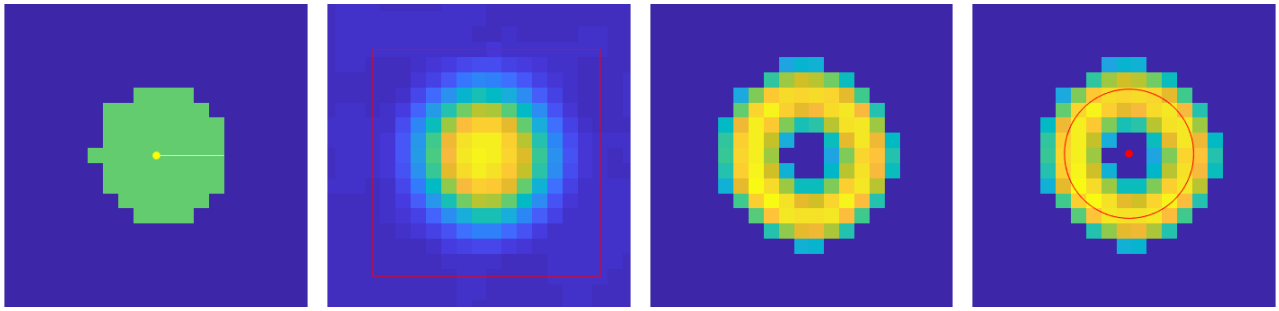
via ∂_x and ∂_y in the two directions, the Total Variation function on $\mathbf{D}$ reads as

$$\mathbf{TV}(\mathbf{D}) = \sqrt{(\partial_x \mathbf{D})^2 + (\partial_y \mathbf{D})^2}. \tag{3.1}$$

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 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.

1. A squared window of interest (WOI) centered in m_k of width $2 \times (1.5 r_k)$ is opened (Figure 3.2(b))



(a) m_k and r_k. (b) WOI. (c) Thresholded **TV**. (d) Estimation.

Fig. 3.2 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.

in **TV** (**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 3.2(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 3.2(d)):

$$\boldsymbol{\alpha} \sim \underset{\alpha_1^2+\alpha_2^2-\alpha_3-a^2\leq 0}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{WR}\boldsymbol{\alpha} - \mathbf{Wy}\|_2^2 + \frac{\mu}{2}\|\boldsymbol{\alpha}\|_2^2 \tag{3.2}$$

where

$$\mathbf{W} = \begin{pmatrix} \sqrt{w_1} & 0 & \dots & 0 \\ 0 & \sqrt{w_2} & \dots & 0 \\ & & & \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & \sqrt{w_q} \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} -2x_1 & -2y_2 & 1 \\ -2x_2 & -2y_2 & 1 \\ -2x_3 & -2y_3 & 1 \\ \vdots & \vdots & \vdots \\ -2x_q & -2y_q & 1 \end{pmatrix}$$

$$\mathbf{y} = -\begin{pmatrix} x_1^2+y_1^2 \\ x_2^2+y_2^2 \\ \vdots \\ x_q^2+y_q^2 \end{pmatrix}, \quad \boldsymbol{\alpha} = \begin{pmatrix} \tilde{\alpha}_1 \\ \tilde{\alpha}_2 \\ \tilde{\alpha}_3 \end{pmatrix}$$

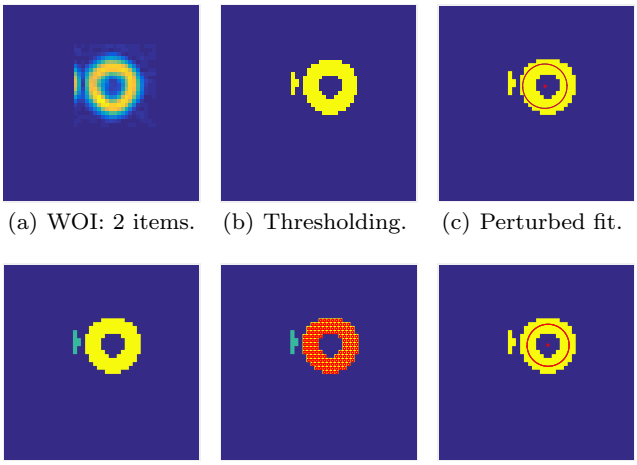
and a is the true radius of the particles. The coordinates of the estimated center (x_k^e, y_k^e) are simply $(\tilde{\alpha}_1, \tilde{\alpha}_2)$, while the estimated radius r_k^e is computed as $r_k^e = \sqrt{\tilde{\alpha}_1^2 + \tilde{\alpha}_2^2 - \tilde{\alpha}_3}$: this is the main reason for the constrain in Equation (3.2).

The regularization term is included due to the fact that the matrix **WR** could be ill–conditioned [21], 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 repeatedly 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 3.3(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 3.3(a) and Figure 3.3(b)), affecting the

least square procedure as it is evident in Figure 3.3(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 3.3(d)), then only the largest one is kept (Figure 3.3(e)). Adopting this procedure leads to a better fit, as shown in Figure 3.3(f).

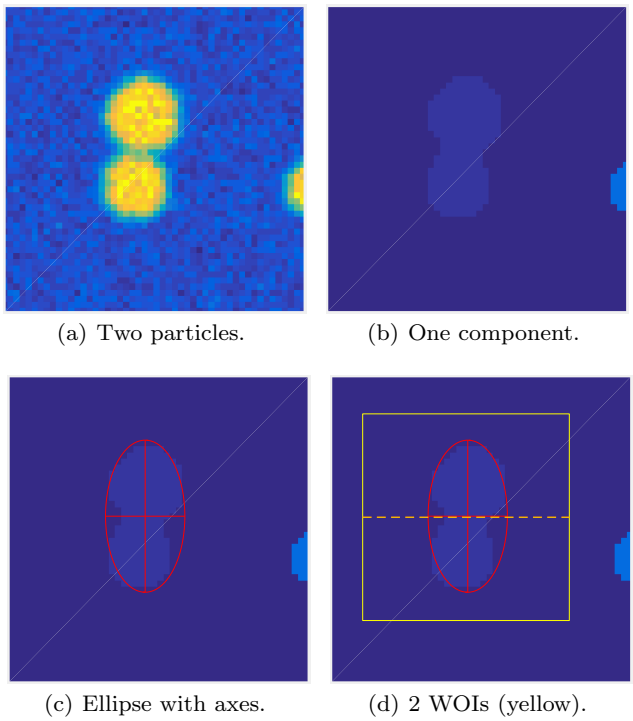


(a) WOI: 2 items. (b) Thresholding. (c) Perturbed fit.

(d) 2 components. (e) Largest item. (f) Improved fit.

Fig. 3.3 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.

Unfortunately, the case in Figure 3.4(a) can occur: the above procedure fails to recognize two distinct particles and compute a center which is very close to 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 [33], in order to be allowed to recognize the different particles.
The second consists of performing a LS fit using an *ellipse model*, instead of a circumference (Figure 3.4(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 the information (length and orientation) of the axes of the ellipse, the WOI can be divided in two smaller WOIs (Figure 3.4(d)): another LS ellipse fit is pursued in each portion. For each one, the ratio of the semi–



(a) Two particles. (b) One component.

(c) Ellipse with axes. (d) 2 WOIs (yellow).

Fig. 3.4 From left to right, ut to bottom: 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.

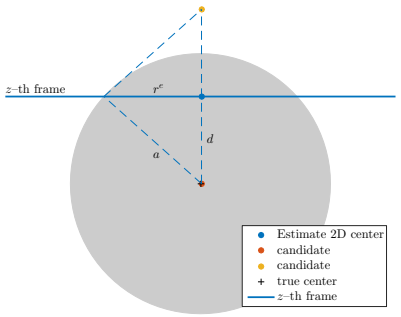
axis is checked again: if it is around 1, then a particle is found, on the other case the same procedure is iterated.

***Remark* 2** *The situation depicted in Figure 3.3 can be worse: 3 or more particles 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 misleading 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 [19, 41, 44].*

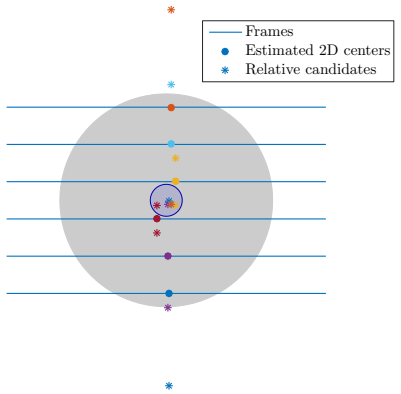
3.2 3–dimensional Estimation

The procedure lying beyond lines 8–9 of Algorithm 1 for the estimation of the center of the particles is now explicited. 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 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 3.5(a)). Hence, the two candidates for the third coordinate are $z\mathrm{d}z - d$ and $z\mathrm{d}z + d$ (with $\mathrm{d}z$ 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 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 3.5(b)). Due to the geometric properties, Z candidates will cluster in a region around the true center (blue enlighten region in Figure 3.5(b)): the next step consists in finding this cluster.



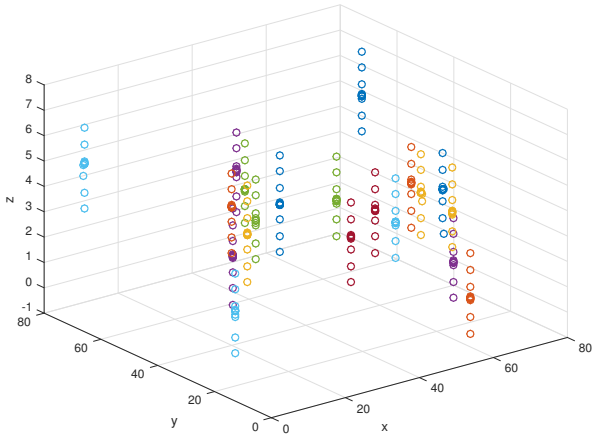
(a) Computation of the center candidates .



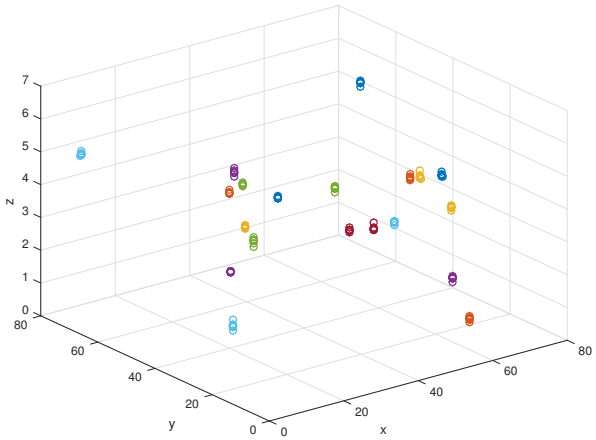
(b) A cluster (in the blue region).

Fig. 3.5 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.

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 3.6 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 3.6(a)). It is expected that there should be a clustering around the true centers of the particles. One strategy could consist of searching for the Z nearest neighbours [28] lying in



(a) Clusters .



(b) Isolated clusters around the centers.

Fig. 3.6 Up: after processing of all the frames of the volume, the clustering of the candidates around the true centers becomes evident. Bottom: 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.

a ball of radius $\rho_{\text{raw}}a$, $0 < \rho_{\text{raw}} < 1$ (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 $\mathcal{R}$;
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 3.7(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 $\rho_{\text{est}}a$, $0 < \rho_{\text{est}} < 1$: let $\left\{\left(x^e_{k,i}, z^e_{k,i}, z^e_{k,i}\right)\right\}_{i=1,\dots,Z}$ be these neighbours (ideally, these are the points lying in the small highlighted circle of Figure 3.5(a)). The estimation of the k–th center $\mathbf{x}^e_k = (x^e_k, y^e_k, z^e_k)$is computed as

$$\begin{aligned} x^e_k &= \frac{1}{A}\sum_{i=1}^{Z} a_i x^e_{k,i} \\ y^e_k &= \frac{1}{A}\sum_{i=1}^{Z} a_i y^e_{k,i} \\ z^e_k &= \frac{1}{A}\sum_{i=1}^{Z} a_i z^e_{k,i} \end{aligned} \tag{3.3}$$

where a_i is the area of the connected component related to the center (x^e_{ki}, y^e_{ki}) (see Subsection 3.1) and $A = \sum_{i=1}^{Z} a_i$. A weighted mean is employed in order to 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 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 ∼20% of a voxel, which is not sufficiently precise in any real–life application, while adopting our proposed procedure yields significantly better results. See Section 4 for the details about error measurements, performance and results.*

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. 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 3μm suspended in a glycerol/water mixture. Both experiments are carried on a MacBookPro, equipped with 16GB RAM 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 [13,35], 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

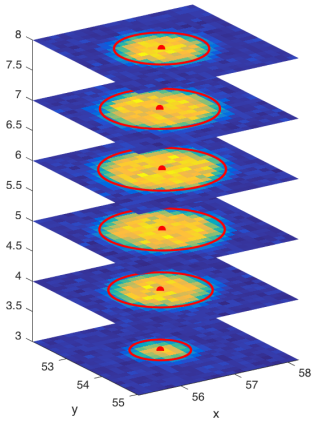
$$\mathrm{T} = \sqrt{(\mathbf{c}-\mathbf{e})^\top \mathbf{D}^{-2} (\mathbf{c}-\mathbf{e})}, \quad \mathbf{D} = \begin{pmatrix} \mathrm{d}x & 0 & 0 \\ 0 & \mathrm{d}y & 0 \\ 0 & 0 & \mathrm{d}z \end{pmatrix} \tag{4.1}$$

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

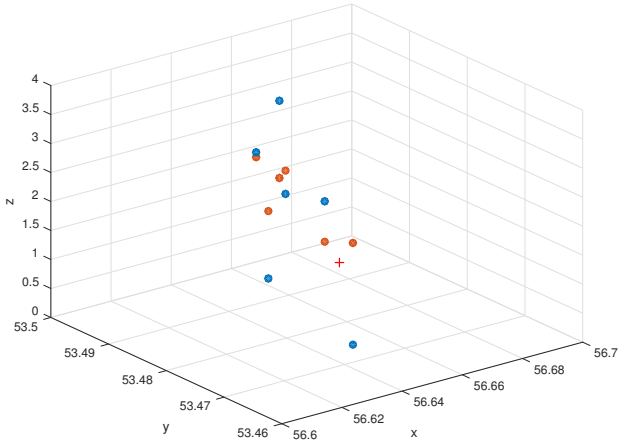
$$\mathrm{P} = \sqrt{\left(\frac{c_x - e_x}{\mathrm{d}x}\right)^2 + \left(\frac{c_y - e_y}{\mathrm{d}y}\right)^2}, \quad \mathrm{V} = \frac{|c_z - e_z|}{\mathrm{d}z}. \tag{4.2}$$

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.

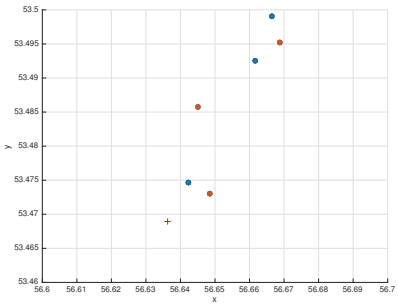
First synthetic test: Gaussian and Poisson noise Following the notation of Section 2, the synthetic dataset is generated using the following settings: $D_x = D_y = 76.8\mu$m, $D_z = 7\mu$m, the number N of particles is 100 of radius $a = 1\mu$m; the volume is discretized into a 3D



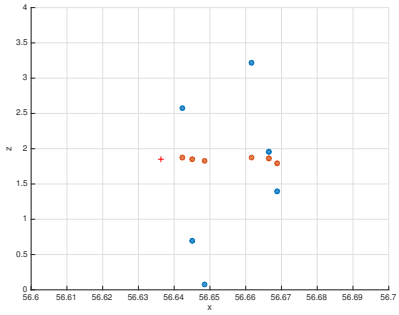
(a) Profiles for LS fit.



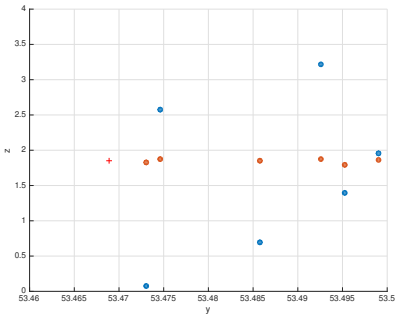
(b) Selected candidates in $\mathcal{C}$ (orange dots).



(c) xy view.



(d) xz view.



(e) yz view.

Fig. 3.7 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 3.7(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 (3.3). 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.

array of dimension $N_x = N_y = 512$, $N_z = 22$, leading to voxels' dimension $\mathrm{d}x = \mathrm{d}y = 0.15$, $\mathrm{d}z = 0.3182$. Two types of noise are affecting the frames: Gaussian ($\sigma_n = 0.2$) and Poisson (see Section 2 for the details on how the Poisson noise is added).

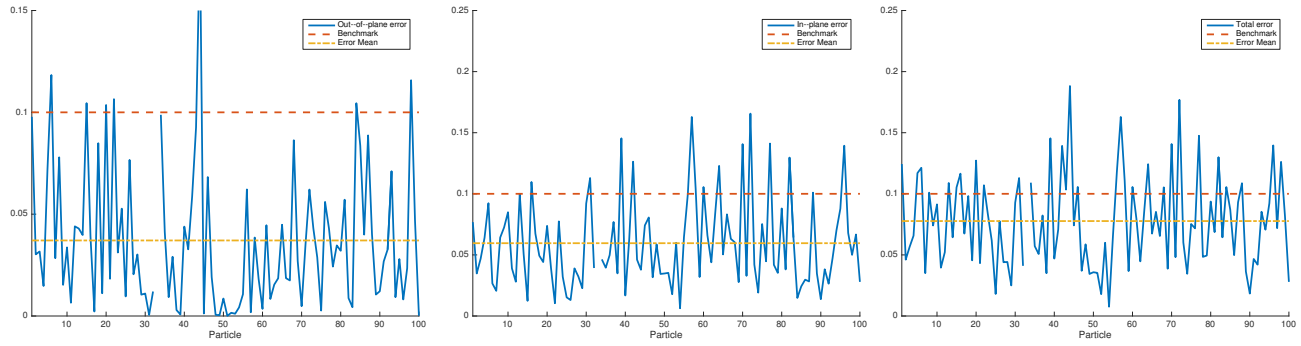
Algorithm 1 is applied: the chosen denoising technique (Line 2) consists simply of filtering via a Gaussian filter of dimension 5 pixels and variance 1. The window of interest is chosen as described in Subsection 3.1. Due to the discretization of the 3D volume, the maximum number Z of frames that can 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 (3.3).

In Figure 4.1 the three type of errors are depicted; the proposed procedure recognizes 99 particles (out of 100). The plots in Figure 4.1 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 [13,15]. In fact, the in–plane error is 0.0596, the out–of–plane error is 0.0371. The total error, given by (4.1), 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 $\mathrm{V}, \mathrm{P}, \mathrm{T}$ for each run. The histograms of the total error T is shown in Figure 4.2(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

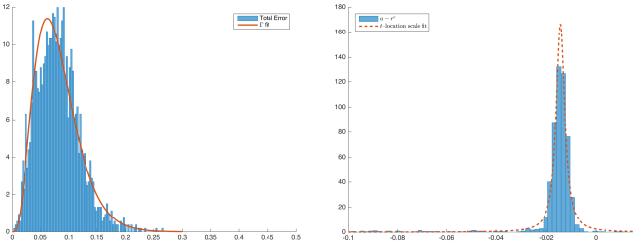
htbp



(a) out–of–plane error. (b) In–plane error. (c) Total error.

Fig. 4.1 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.

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% [13]. Our proposed procedure is based



(a) Distribution of T. (b) Distribution of $a - r^e$.

Fig. 4.2 (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.

on the assumption that the true radius 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 4.2(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 4.2(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.

The last part is devoted to study the performance w.r.t. the vertical resolution, 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 Section 4 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 done: the numbers appearing in Section 4 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 -0.013, meaning that, regardless the number of vertical frames, the radius of the particles is overestimated by 1.3%. The last line of Section 4 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 to have a low number of candidates in $\mathcal{C}$. Hence, it is a problem linked to the relation between the dimension

Table 4.1 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.

	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

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 ($a = 1.1\mu$m) 10 frames prove to be sufficient, as it is evident in Table 4.2)

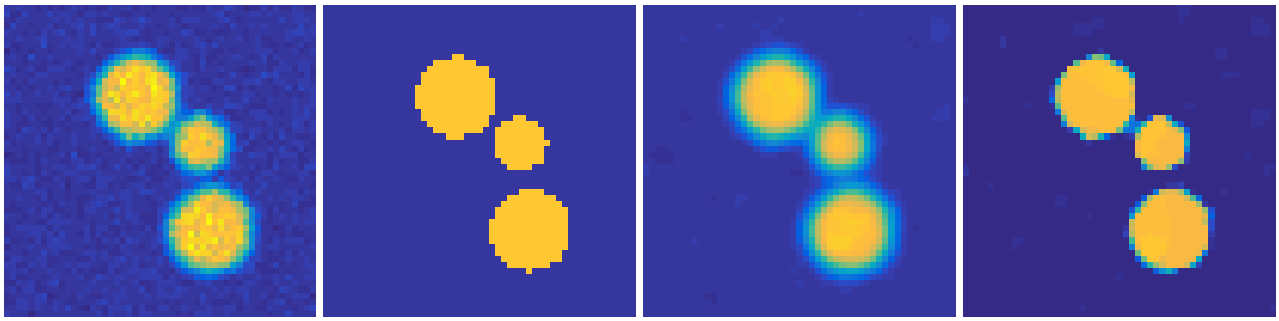
Table 4.2 Results of 20 runs of the procedure with $a = 1.1\mu$m, $N_z = 10$ and $N = 50$. It is evident that the poor performance of the procedure when $N_z = 10$ in Section 4 is due to the relation between the diameter of the particles and the resolution. Such a low resolution is however enough for slightly larger particles to get reliable results.

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

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: $D_x = D_y = 76.8\mu$m, $D_z = 7\mu$m, 100 particles of radius $a = 1\mu$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 affects the data. Algorithm 1 is applied to this dataset: satisfactory results, in line with the ones in Section 4, 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 [6]: 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 [6] has been chosen instead of possibly simpler procedures (e.g., [9,10,12,16]) for its ability to increase contrast [7,3,8] 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 4.3, where a zoom of the 4–th frame is shown. The Bregman procedure uses as inner solver the AEM algorithm [11], with a maximum of 1000 iterations maximum and stopped via the criterion described in [6] 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: 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 not sufficient to support this claim, further tests are carried on and presented in Table 4.3: 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 1.1μm. The second test is performed on a dataset with the same characteristic of the first one presented in this paragraph: $D_x = D_y = 76.8\mu$m, $D_z = 7\mu$m, 100 particles of radius $a = 1\mu$m, $N_x = N_y = 512$, $N_z = 22$.

Table 4.3 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%,



(a) Blurred & noisy. (b) Clear image. (c) Gaussian filtering. (d) Bregman.

Fig. 4.3 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 4.3 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 4.2(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 Gaussin 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%.

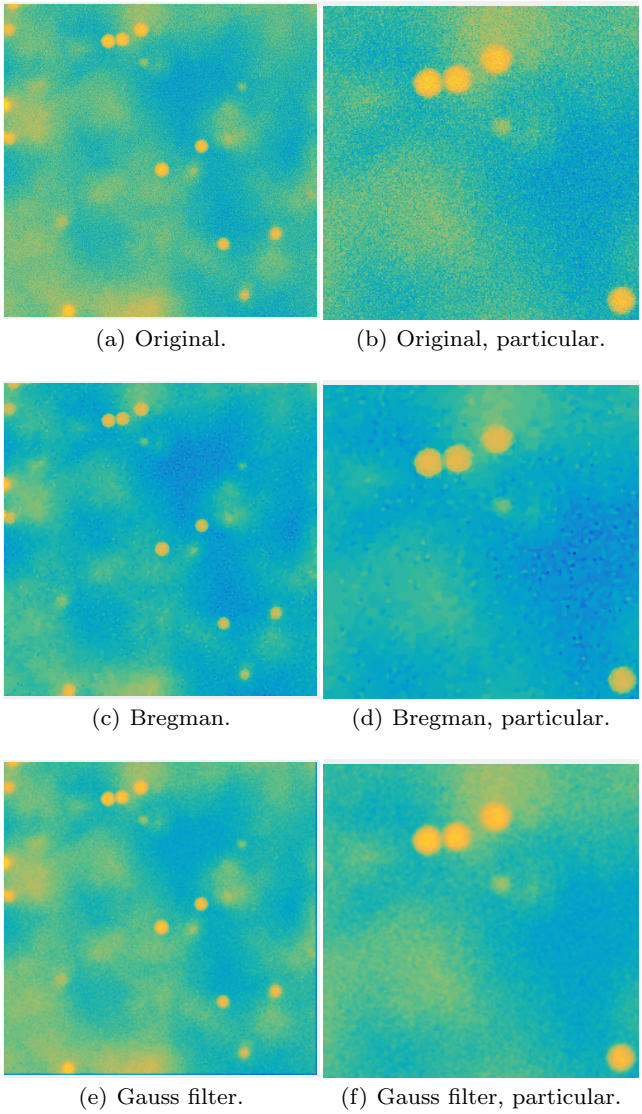
N_z	Technique	P	V	T	$a - r^e$	$N_{\text{rec}}(\%)$
10	Gaussian Filter	0.0622	0.0271	0.0712	-0.0096	46.7 (93.4%)
	Bregman	0.0628	0.0177	0.0677	-0.0013	47.6 (95.2%)
22	Gaussian Filter	0.0626	0.0380	0.0793	-0.0138	98.8 (98.8%)
	Bregman	0.0637	0.0291	0.0746	-0.0019	99.4 (99.4%)

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 techinque is true: at a first sight, it seems from Section 4 that this is a *determinate error* [13] 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 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 $D_x = D_y = 64\mu$m, $D_z = 4.1\mu$m, discretized into an array of dimension $512 \times 512 \times 10$, leading to $\mathrm{d}x = \mathrm{d}y = 0.125\mu$m, $\mathrm{d}z = 0.41\mu$m; 50 scans of the volume were recorded, with a $\mathrm{d}t = 0.5$s. The diameter of the particles is 3μm ($a = 1.5\mu$m) and they are suspended in a $\sim$ 70%–30% glycerol/water mixture (viscosity of ~ 0.017 Pa s). The instrument used to acquire this data is a confocal microscope (Zeiss LSM 700) with a 100×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 tolerance of 10^{-4} for the stopping criterion described in [6] with

$\alpha = 2, 3$ external iteration are allowed. Since the images are given without any information about their recording, a Gaussian PSF[1] with $\sigma = 1$ and zero mean is assumed as blurring 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. Figure 4.4 shows in its first row the 6–th acquired frame at time $t = 1$, the restored 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.



(a) Original. (b) Original, particular.

(c) Bregman. (d) Bregman, particular.

(e) Gauss filter. (f) Gauss filter, particular.

Fig. 4.4 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 $[0, 255]$.

[1] `fspecial('gaussian',512,1)`.

Algorithm 1 is set with an initial WOI of width $2 \times 0.1 r_k$ (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 4.5(b)–Figure 4.5(k).

Figure 4.5(a) provides a visual inspection of the reconstructed position of the 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 from Figure 4.5(b) to Figure 4.5(k), where the recovered profiles of the particles are superimposed on the original images. In these images, the top left corner corresponds to the point $(0, 0, k\mathrm{d}z)$ in the 3D space, being k the number of the frame. A closer inspection of Figure 4.5 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.

5 Conclusion

In this work, a particle segmentation and position estimation methodology is presented. Assuming fixed spherical particles with a known radius, this procedure on the first hand applies a noise removal algorithm on each frame of the 3D volume, then it 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 a voxel, which is the state–of–the–art performance of this type of algorithms. Furthermore, the intermediate steps implemented for the 3D reconstruction allow also to recover the particles' position within each 2D frame, with a subpixel precision. Reliable results for the 3D positioning are achieved even for a low vertical resolution: the total error is indeed under the 10% of a voxel. Moreover, the very low error on the radius estimation suggests that this procedure improve *a priori* information about the radius of particles of uncertain dimension. This work demonstrate that the preprocessing of the frames requires particularly tailored techniques, depending on noise type: since Poisson

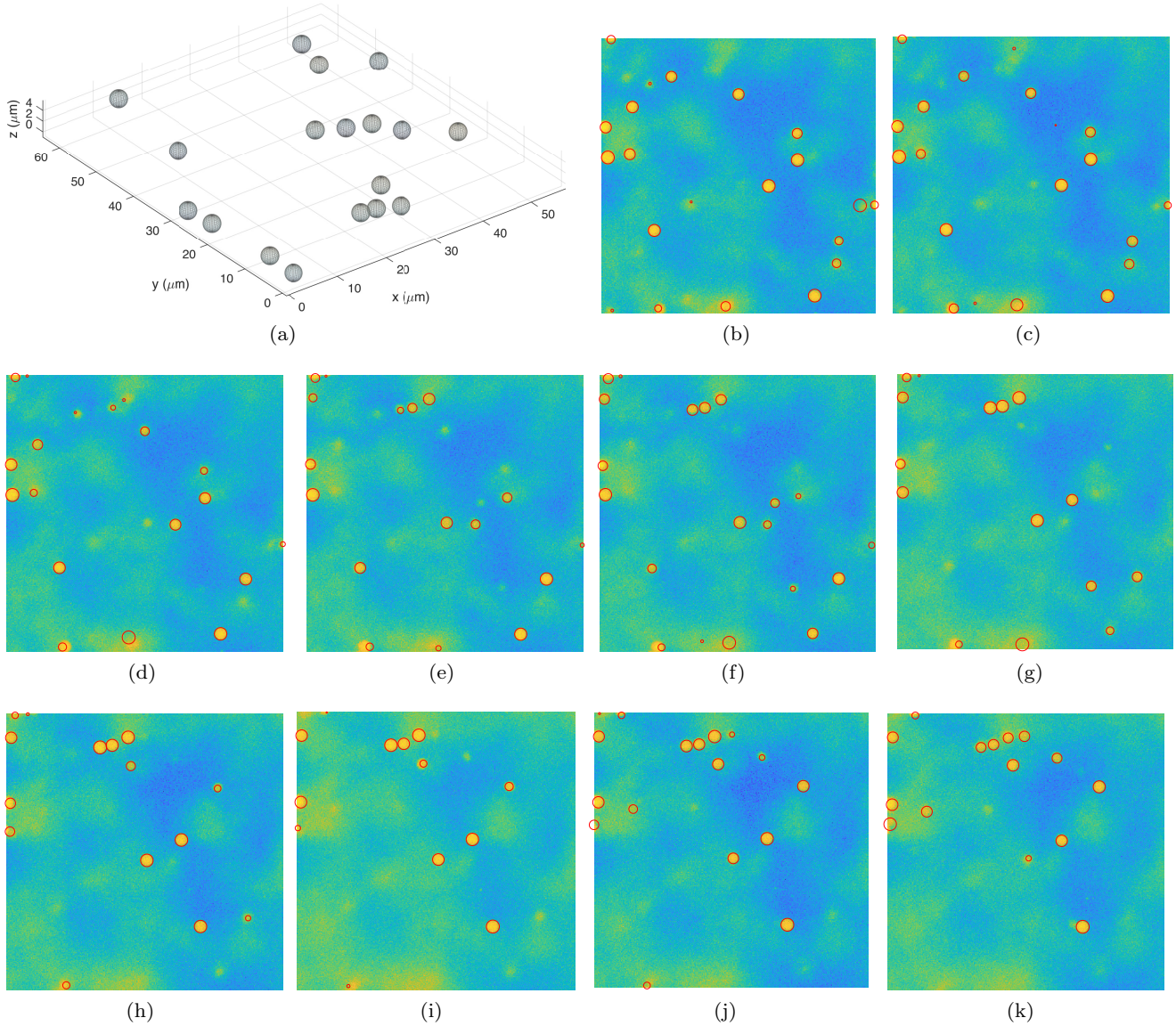


Fig. 4.5 Panel (a) 3D recovering of the position of the particles at time $t = 1$.. Panel from (b) to (k) contain the original images with the superposition of the recognized profiles (in red).

noise is the most common noise affecting the images, simple Gaussian filtering is not sufficient. One of the available image restoration techniques is then applied in this context: although they are more demanding in term of computational cost and time, the application of this strategies leads to a general improvement of the position estimation. Moreover, this tailored approach significantly increases the precision on the radius estimation, and it provide deeper insights on the role of Gaussian filtering in this task, proving that it induces an overestimation. Future work will involve better segmentation techniques for pathological cases, employing more tailored approaches such as regularized approaches inspired by the Mumford-Shah functional. The case of spherical particles with unknown radius will be also handled. The reliable results in positioning directly suggest that the proposed technique can be embedded in a more general procedure devoted to tracking procedure, where the particles are no longer fixed but may subjected to significant Brownian motion between slice acquisition.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Akhmanova, A., Steinmetz, M.O.: Tracking the ends: a dynamic protein network controls the fate of microtubule tips. Nature Reviews Molecular Cell Biology **9**, 309 EP – (2008)
2. Apgar, J., Tseng, Y., Fedorov, E., Herwig, M.B., Almo, S.C., Wirtz, D.: Multiple-particle tracking measurements of heterogeneities in solutions of actin filaments and actin bundles. Biophysical Journal **79**(2), 1095–1106 (2000)
3. Benfenati, A., Camera, A.L., Carbillet, M.: Deconvolution of post-adaptive optics images of faint circumstellar environments by means of the inexact Bregman procedure. Astronomy & Astrophysics **586**, A16 (2016). DOI 10.1051/0004-6361/201526960
4. Benfenati, A., Coscia, V.: Nonlinear microscale interactions in the kinetic theory of active particles. Applied Mathematics Letters **26**(10), 979 – 983 (2013). DOI https://doi.org/10.1016/j.aml.2013.04.007
5. Benfenati, A., Coscia, V.: Modeling opinion formation in the kinetic theory of active particles i: spontaneous trend. ANNALI DELL'UNIVERSITA' DI FERRARA **60**(1), 35–53 (2014). DOI 10.1007/s11565-014-0207-2
6. Benfenati, A., Ruggiero, V.: Inexact Bregman iteration with an application to Poisson data reconstruction. Inverse Problems **29**(6), 065016 (2013)
7. Benfenati, A., Ruggiero, V.: Image regularization for Poisson data. Journal of Physics: Conference Series **657**(1), 012011 (2015)
8. Benfenati, A., Ruggiero, V.: Inexact Bregman iteration for deconvolution of superimposed extended and point sources. Communications in Nonlinear Science and Numerical Simulaion **21**(1), 210–224 (2015)
9. Bonettini, S., Benfenati, A., Ruggiero, V.: Primal-dual first order methods for total variation image restoration in presence of Poisson noise. In: 2014 IEEE International Conference on Image Processing (ICIP), pp. 4156–4160 (2014). DOI 10.1109/ICIP.2014.7025844
10. Bonettini, S., Benfenati, A., Ruggiero, V.: Scaling techniques for ε – subgradient methods. SIAM Journal on Optimization **26**(3), 1741–1772 (2016). DOI 10.1137/14097642X
11. Bonettini, S., Ruggiero, V.: An alternating extragradient method for total variation-based image restoration from poisson data. Inverse Problems **27**(9), 095001 (2011)
12. Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. SIAM Journal on Imaging Sciences **3**(3), 492–526 (2010). DOI 10.1137/090769521. URL `https://doi.org/10.1137/090769521`
13. Cheezum, M.K., Walker, W.F., Guilford, W.H.: Quantitative comparison of algorithms for tracking single fluorescent particles. Biophysical Journal **81**(4), 2378–2388 (2001)
14. Chen, D.T., Weeks, E.R., Crocker, J.C., Islam, M.F., Verma, R., Gruber, J., Levine, A.J., Lubensky, T.C., Yodh, A.G.: Rheological microscopy: Local mechanical properties from microrheology. Phys. Rev. Lett. **90**, 108301 (2003). DOI 10.1103/PhysRevLett.90.108301
15. Chenouard, N., Smal, I., de Chaumont, F., Maška, M., Sbalzarini, I.F., Gong, Y., Cardinale, J., Carthel, C., Coraluppi, S., Winter, M., Cohen, A.R., Godinez, W.J., Rohr, K., Kalaidzidis, Y., Liang, L., Duncan, J., Shen, H., Xu, Y., Magnusson, K.E.G., Jaldén, J., Blau, H.M., Paul-Gilloteaux, P., Roudot, P., Kervrann, C., Waharte, F., Tinevez, J.Y., Shorte, S.L., Willemse, J., Celler, K., van Wezel, G.P., Dan, H.W., Tsai, Y.S., de Solórzano, C.O., Olivo-Marin, J.C., Meijering, E.: Objective comparison of particle tracking methods. Nature Methods **11**, 281 EP – (2014)
16. Chouzenoux, E., Jezierska, A., Pesquet, J., Talbot, H.: A convex approach for image restoration with exact poisson–gaussian likelihood. SIAM Journal on Imaging Sciences **8**(4), 2662–2682 (2015). DOI 10.1137/15M1014395. URL `https://doi.org/10.1137/15M1014395`
17. Chu, K.K., Mojahed, D., Fernandez, C.M., Li, Y., Liu, L., Wilsterman, E.J., Diephuis, B., Birket, S.E., Bowers, H., Solomon, G.M., Schuster, B.S., Hanes, J., Rowe, S.M., Tearney, G.J.: Particle-tracking microrheology using micro-optical coherence tomography. Biophysical Journal **111**(5), 1053 – 1063 (2016). DOI https://doi.org/10.1016/j.bpj.2016.07.020
18. Figueiredo, M.A.T., Bioucas-Dias, J.M.: Deconvolution of Poissonian images using variable augmented Lagrangian optimization. IEEE Workshop on Statistical Signal Processing, Cardiff (2009)
19. Foare, M., Lachaud, J.O., Talbot, H.: Image restoration and segmentation using the Ambrosio–Tortorelli functional and discrete calculus. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 1418–1423 (2016). DOI 10.1109/ICPR.2016.7899836
20. Godinez, W.J., Rohr, K.: Tracking multiple particles in fluorescence time-lapse microscopy images via probabilistic data association. IEEE Transactions on Medical Imaging **34**(2), 415–432 (2015). DOI 10.1109/TMI.2014.2359541
21. Golub, G., Van Loan, C.: Matrix Computations. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press (2013)
22. Hansen, P.C., Nagy, J.G., O'Learly, D.P.: Deblurring images: matrices, spectra, and filtering. SIAM (2006)
23. Husain, M., Boudier, T., Paul-Gilloteaux, P., Casuso, I., Scheuring, S.: Software for drift compensation, particle tracking and particle analysis of high–speed atomic force microscopy image series. Journal of Molecular Recognition **25**(5), 292–298 (2012). DOI 10.1002/jmr.2187
24. Jandt, U., Zeng, A.P.: Modeling of Intracellular Transport and Compartmentation, pp. 221–249. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). DOI 10.1007/10_2011_104
25. Jenkins, M., Egelhaaf, S.: Confocal microscopy of colloidal particles: Towards reliable, optimum coordinates. Advances in Colloid and Interface Science **136**(1), 65 – 92 (2008). DOI 10.1016/j.cis.2007.07.006
26. Josephson, L.L., Swan, J.W., Furst, E.M.: In situ measurement of localization error in particle tracking microrheology. Rheologica Acta (2018). DOI 10.1007/s00397-018-1117-5
27. Kodippili, G.C., Putt, K.S., Low, P.S.: Evidence for three populations of the glucose transporter in the human erythrocyte membrane. Blood Cells, Molecules, and Diseases **77**, 61 – 66 (2019). DOI https://doi.org/10.1016/j.bcmd.2019.03.005
28. Kononenko, I., Kukar, M.: Machine Learning and Data Mining: Introduction to Principles and Algorithms. Horwood Publishing Limited (2007)
29. Kusumi, A., Tsunoyama, T.A., Hirosawa, K.M., Kasai, R.S., Fujiwara, T.K.: Tracking single molecules at work in living cells. Nature Chemical Biology **10**, 524 EP – (2014). DOI 10.1038/nchembio.1558
30. Lin, T.S., Zhu, S., Kojima, S., Homma, M., Lo, C.J.: Flil association with flagellar stator in the sodium-driven

vibrio motor characterized by the fluorescent microscopy. Scientific reports **8**(1), 11172 (2018)
31. Magnusson, K.E.G., Jaldén, J.: 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), pp. 382–385 (2012). DOI 10.1109/ISBI.2012.6235564
32. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics **9**(1), 62–66 (1979). DOI 10.1109/TSMC.1979.4310076
33. Puybareau, É., Talbot, H., Gaber, N., Bourouina, T.: 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, pp. 486–497. Springer International Publishing, Cham (2017)
34. Savin, T., Doyle, P.S.: Static and dynamic errors in particle tracking microrheology. Biophysical Journal **88**(1), 623 – 638 (2005). DOI https://doi.org/10.1529/biophysj.104.042457
35. Savin, T., Doyle, P.S.: Static and dynamic errors in particle tracking microrheology. Biophysical Journal **88**(1), 623 – 638 (2005). DOI 10.1529/biophysj.104.042457
36. Sbalzarini, I., Koumoutsakos, P.: Feature point tracking and trajectory analysis for video imaging in cell biology. Journal of Structural Biology **151**(2), 182 – 195 (2005). DOI 10.1016/j.jsb.2005.06.002
37. Valentine, M.T., Kaplan, P.D., Thota, D., Crocker, J.C., Gisler, T., Prud'homme, R.K., Beck, M., Weitz, D.A.: Investigating the microenvironments of inhomogeneous soft materials with multiple particle tracking. Phys. Rev. E **64**, 061506 (2001). DOI 10.1103/PhysRevE.64.061506
38. Wagner, C.E., Turner, B.S., Rubinstein, M., McKinley, G.H., Ribbeck, K.: A rheological study of the association and dynamics of muc5ac gels. Biomacromolecules **18**(11), 3654–3664 (2017). DOI 10.1021/acs.biomac.7b00809. PMID: 28903557
39. Wen, L., Zheng, Z.H., Liu, A.A., Lv, C., Zhang, L.J., Ao, J., Zhang, Z.L., Wang, H.Z., Lin, Y., Pang, D.W.: Tracking single baculovirus retrograde transportation in host cell via quantum dot-labeling of virus internal component. Journal of Nanobiotechnology **15**(1), 37 (2017). DOI 10.1186/s12951-017-0270-9. URL `https://doi.org/10.1186/s12951-017-0270-9`
40. Xu, Q., Boylan, N., Suk, J., Wang, Y., Nance, E., Yang, J., McDonnell, P., Cone, R., Duh, E., Hanes, J.: Nanoparticle diffusion in, and microrheology of, the bovine vitreous ex vivo. Journal of Controlled Release **167**(1), 76–84 (2013). DOI 10.1016/j.jconrel.2013.01.018
41. Yap, C.K., Lee, H.K.: Identification of cell nucleus using a Mumford-Shah ellipse detector. In: Advances in Visual Computing, pp. 582–593. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
42. Yildiz, A., Forkey, J.N., McKinney, S.A., Ha, T., Goldman, Y.E., Selvin, P.R.: Myosin V walks hand–over–hand: single fluorophore imaging with 1.5–nm localization. Science **300**(5628), 2061–2065 (2003)
43. Zanella, R., Boccacci, P., Zanni, L., Bertero, M.: Efficient gradient projection methods for edge-preserving removal of Poisson noise. Inverse Problems **25**, 045010 (2009)
44. Zanella, R., Porta, F., Ruggiero, V., Zanetti, M.: Serial and parallel approaches for image segmentation by numerical minimization of a second-order functional. Appl. Math. Comput. **318**(C), 153–175 (2018). DOI 10.1016/j.amc.2017.07.021