Tracking the dynamics of translation and absolute orientation of a sphere in a turbulent flow

Robert Zimmermann,[1] Yoann Gasteuil,[1] Mickael Bourgoin,[1,2] Romain Volk,[1] Alain Pumir,[1] and Jean-François Pinton[1]
(International Collaboration for Turbulence)
[1]*Laboratoire de Physique, CNR, UMR 5672, Ecole Normale Supérieure de Lyon, Lyon F-69007, France*
[2]*Laboratoire des Ecoulements Géophysiques et Industriels, CNRS, UJF INPG, Grenoble, F-38041, France*

We study the six-dimensional dynamics—position and orientation—of a large sphere advected by a turbulent flow. The movement of the sphere is recorded with two high-speed cameras. Its orientation is tracked using a novel, efficient algorithm; it is based on the identification of possible orientation "candidates" at each time step, with the dynamics later obtained from maximization of a likelihood function. Analysis of the resulting linear and angular velocities and accelerations reveal a surprising intermittency for an object whose size lies in the inertial range, close to the integral scale of the underlying turbulent flow. © *2011 American Institute of Physics*. [doi:10.1063/1.3554304]

I. INTRODUCTION

The advent of resolved Lagrangian measurements has helped understand the dynamics of turbulence from the point of view of fluid particles.[1] In the experiments, solid *tracers* are followed in lieu of fluid particles, which naturally raises the question of the understanding of the dynamics of *finite size* objects in turbulent flows. It is a subclass of the issue of the dynamics of *inertial particles*, *i.e.*, particles which have inertia with respect to the fluid motions, either because their density differs from that of the fluid or because their spatial extent cannot be ignored. If the particles are quite small compared to the smallest fluid motion (the Kolmogorov dissipative length scale η), arguments show that they behave as tracers of fluid motions. Observations have revealed a very intense intermittency in the motion of fluid tracers.[2,3] They experience very strong accelerations, with a probability distribution which displays stretched exponential tails.[4]

When the diameter D of the advected particles is of the order of, or larger than η, their equation of motion is not known (see, however Refs. 5–7). We restrict our discussion to neutrally buoyant spheres. Several recent studies[8–11] have shown that the acceleration statistics of such inertial particles does not gently reduce to a Gaussian behavior as their diameter increases. It is an important feature because the characterization of forces acting on an object advected by a turbulent flow has many applications in engineering (from mixing issues in industrial processes to dispersion in the oceans or in the atmosphere).

The study reported here takes a leap forward in size and considers the motion of a neutrally buoyant sphere with diameter D of the order of the integral scale L_{int} (the scale at which energy is fed into the flow). In addition, we aim at resolving the 6 degrees of freedom of the particle dynamics, *i.e.*, the goal is to obtain a simultaneous tracking in time of the particle's position and its absolute orientation with respect to a reference frame. This enables the study of the forces and torques acting on a large inertial particle, thus permitting to ask yet fundamental questions about their dynamics.[12] Furthermore, the absolute orientation is important for future work using instrumented particles and for problems where there is a preferential direction—such as when there is a global rotation, a temperature gradient, or an imposed magnetic field.

The tracking of the particle position in space can be carried out by using methods already developed and successfully tested for small particles.[13] In comparison, following the *absolute* orientation of the particle is much more challenging, both because of the specifics of angular variables, and of specific algorithmic requirements.

Previous studies focused on directly measuring the angular velocity without resolving the absolute orientation as a function of time. Ye and Roco[14] tracked dots painted on a particle with high-speed cameras and computed the angular velocity from their displacement between two consecutive frames. Frish and Webb[15] created an Eulerian technique measuring one component of the angular velocity using specially engineered, transparent particles which contain an embedded mirror. The reported particle diameter is less than $50\,\mu m$, which is of the order of the Kolmogorov length scale, η.

The principle used here is completely different: it consists simply in painting the particle with a suitable layout, and in retrieving its orientation. For algorithmic efficiency (and robustness) this is not done step by step but for the entire trajectory using a global path extraction.

The text below is organized as follows: we first present the experimental setup and recall some important features of the orientation algebra in 3D. We then describe how the particle images are extracted from the movie images, and compared to synthetic images with arbitrary orientations. Possible candidates are identified and then assembled into an orientation time series using a global maximization of a likelihood function. Finally, we present some results concerning the particle dynamics.

82, 033906-1

II. BASICS

A. Experimental setup

A turbulent flow is generated in the gap between two counter-rotating impellers of radius $R = 10$ cm fitted with straight blades 1 cm in height. The flow domain in between the impeller has characteristic lengths $H = 2R = 20$ cm and the working fluid is a water–glycerol mixture, whose density can be finely tuned. In order to be able to perform direct optical measurements, the container is build with flat Plexiglas (Poly[methyl methacrylate]) side walls, so that the cross section of the vessel is square. This type of von Kármán swirling flow has been used extensively in the past for the study of fully developed turbulence;[1] its local characteristics approximate homogeneous turbulence in its center, although it is known to have a large scale anisotropy.[16,17] A sketch of the setup is provided in Fig. 1—further details about the flow turbulence are given later in Sec. IV.

A white, PolyAmid sphere with diameter $D = 18$ mm (accuracy 0.01 mm, Marteau & Lemarié, France) moves and rotates in the turbulent flow. It is neutrally buoyant in the fluid—whose density is adjusted to that of the particle $\rho_p = 1.14$ g cm^{-3} by addition of glycerol to water. The density mismatch, measured from sedimentation speeds, is found to be less than $\Delta\rho / \rho = 10^{-4}$. The particle is textured black and white by hand using either black nail polish or a black-ink permanent marker. Its motion is tracked using two high-speed video cameras (Phantom V12, Vision Research) which record synchronously two views at $\sim 90°$. The flow is illuminated by high power light-emitting diodes (LEDs) and sequences of 8 bit gray scale images are recorded at a rate of 600 frames per second.

Both cameras observe the measurement region with a resolution of 650×650 pixels, covering a volume of $15 \times 15 \times 15$ cm^3. Hence, the particle diameter is 70–90 pixels. In the choice of the particle texture, several features have to be considered:

- A single view should correspond to a unique orientation.

- Illumination inhomogeneities may cause regions to look similar in the camera images. Optically resembling views should correspond to clearly distinct orientations.

- The cameras are gray scale so the texture has to be black and white.

- The number of black and white pixel should be approximately the same in every possible view.

In our configuration, the camera can store on the order of 15 000 frames in on-board memory, thus limiting the duration of continuous tracks. The movies are downloaded to a personal computer (PC), waiting to be processed. The processing is done on a gaming PC with a state of the art graphics card. Algorithm development and code testing is done on an Apple Macbook Pro. The code is written in MATLAB 2009a using the image and signal processing toolboxes as well as the PSYCHTOOLBOX extensions[18,19] which provide OPENGL wrappers for MATLAB.

B. Angular variables

The parametrization of an angular position in 3D space causes a number of difficulties which are briefly addressed in this section (see, e.g., Refs. 20–22 for a more complete presentation). One of them is caused by the degeneracy of the axes of rotation for certain orientations (the "gimbal lock" problem). Another is the choice of a suitable measure of distance between two orientations.

1. Describing orientations

As stated by the Euler rotation theorem, three parameters are needed to describe any rotation in 3D. We use here Euler angles with the Tait–Bryan convention as shown in Fig. 2. In the transformation from lab to particle coordinate system, we first apply a rotation around the z axis of angle θ_z, followed by a rotation around the intermediate y axis of angle θ_y and last a rotation of angle θ_x around the new x axis. The rotations work on the object using a right handed coordinate system and right handed direction of rotation. We will denote an orientation triplet by an underscore, *e.g.*, $\underline{\theta}$, in order to distinguish them from vectors (which are typeset in bold font, *e.g.*, $\boldsymbol{\omega}$).

The orientation of the object is fully described by an orthogonal 3×3 matrix $\underline{\underline{R}}$, obtained from the composition of
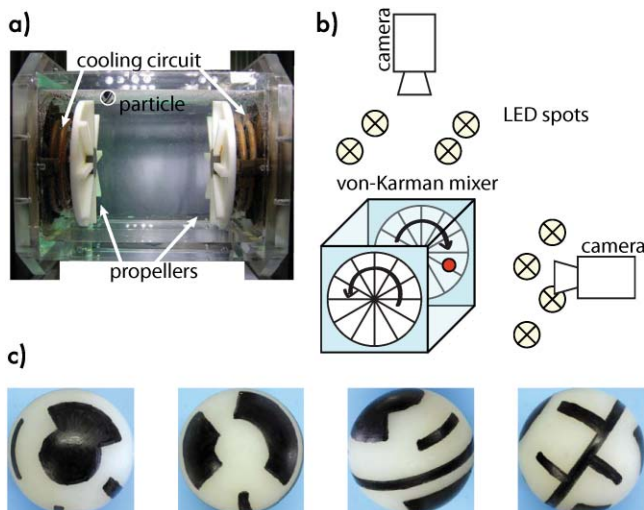

FIG. 1. (Color online) Sketch of the experimental setup: (a) image of the von Kármán mixer, (b) sketch of the camera arrangement, and (c) textured sphere for different orientations.
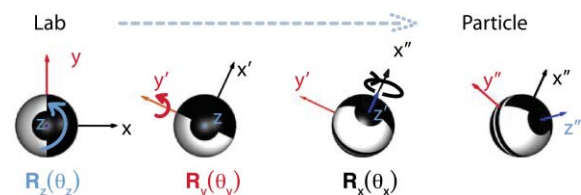

FIG. 2. (Color online) Tait–Bryan rotation sequence describing the sphere's orientation.

the three elementary rotations

$$\underline{\underline{R}}(\theta_x, \theta_y, \theta_z) = \underline{\underline{R}}_x(\theta_x)\underline{\underline{R}}_y(\theta_y)\underline{\underline{R}}_z(\theta_z)$$

$$= \begin{bmatrix} c\theta_y\,c\theta_z & -c\theta_y\,s\theta_z & s\theta_y \\ s\theta_x\,s\theta_y\,c\theta_z + c\theta_x\,s\theta_z & -s\theta_x\,s\theta_y\,s\theta_z + c\theta_x\,c\theta_z & -s\theta_x\,c\theta_y \\ -c\theta_x\,s\theta_y\,c\theta_z + s\theta_x\,s\theta_z & c\theta_x\,s\theta_y\,s\theta_z + s\theta_x\,c\theta_z & c\theta_x\,c\theta_y \end{bmatrix}$$

$$(1)$$

with $c\cdot = \cos(\cdot)$ and $s\cdot = \sin(\cdot)$. Consequently, from any rotation matrix the three Euler angles can be extracted using

$$\underline{\theta} = \{\theta_x, \theta_y, \theta_z\} = \begin{pmatrix} \mathrm{atan2}(-\underline{\underline{R}}_{23}, \underline{\underline{R}}_{33}) \\ \mathrm{asin}(\underline{\underline{R}}_{13}) \\ \mathrm{atan2}(-\underline{\underline{R}}_{12}, \underline{\underline{R}}_{11}) \end{pmatrix} \qquad (2)$$

enforcing $\theta_x, \theta_z \in [0, 2\pi[$ and $\theta_y \in [-\pi/2, \pi/2]$. However this choice is not unique because there is a second triplet with $\underline{\underline{R}}(\theta_x + \pi, \mathrm{sign}(\theta_y)\cdot\pi - \theta_y, \theta_z + \pi) = \underline{\underline{R}}(\theta_x, \theta_y, \theta_z)$. Needless to say, multiples of 2π can be added to each angle. An important practical consequence is that even for small changes in orientation the difference between two Euler angle triplets, $\underline{\theta}_1$ and $\underline{\theta}_2$, has formally four possible results.

The curvilinear coordinate $\underline{\theta}$ is related to the angular velocity, $\boldsymbol{\omega}^{\mathbb{P}}$, (in the particle frame) by

$$\boldsymbol{\omega}^{\mathbb{P}}\big(\underline{\theta}(t)\big) = \begin{bmatrix} 1 & 0 & s\theta_y \\ 0 & c\theta_x & -s\theta_x\,c\theta_y \\ 0 & s\theta_x & c\theta_x\,c\theta_y \end{bmatrix} \cdot \frac{d}{dt}\begin{pmatrix} \theta_x \\ \theta_y \\ \theta_z \end{pmatrix}$$

$$= \underline{\underline{H}}(\theta_x, \theta_y) \cdot \frac{d}{dt}\begin{pmatrix} \theta_x \\ \theta_y \\ \theta_z \end{pmatrix}. \qquad (3)$$

For $\cos(\theta_y) \approx 0$, the determinant of the matrix $\underline{\underline{H}}$, $\det(\underline{\underline{H}})$, vanishes and its inverse is not defined. In other words, finite body rotations need infinite change in the Euler angles. This singularity is called a *gimbal lock* and is a well-known problem in robotics and aerospace engineering. Geometrically, the second rotation turns the first axis parallel to the third axis of rotation, and the rotation loses 2 degrees of freedom. Unfortunately gimbal locks cannot be avoided by a wise choice of representation.

One then needs to define a distance between two arbitrary orientations, immune to this type of singularity. A natural distance between two arbitrary orientation matrixes, $\underline{\underline{A}}$ and $\underline{\underline{B}}$, is

$$\mathrm{Tr}\big((\underline{\underline{A}} - \underline{\underline{B}})(\underline{\underline{A}} - \underline{\underline{B}})^T\big) = 6 - 2\mathrm{Tr}\big(\underline{\underline{A}}\underline{\underline{B}}^T\big)$$

$$= 4(1 - \cos(\phi)) \qquad (4)$$

using $\underline{\underline{A}}\,\underline{\underline{A}}^T = \underline{\underline{B}}\,\underline{\underline{B}}^T = 1$ and that $\underline{\underline{A}}\underline{\underline{B}}^T$ is a rotation matrix with the eigenvalues $1, e^{i\phi}, e^{-i\phi}$. The distance is thus a growing function of ϕ. We measure here the distance between two rotation matrices by

$$d(\underline{\underline{A}}, \underline{\underline{B}}) \equiv \mathrm{acos}\left(\frac{1}{2}\big[\mathrm{Tr}\big(\underline{\underline{A}}\,\underline{\underline{B}}^T\big) - 1\big]\right). \qquad (5)$$

Because it works directly on the orientation matrices it is neither sensitive to gimbal locks nor to the choice of the representation and thus an important tool in our algorithm. It should be noted that $d(\underline{\underline{A}}, \underline{\underline{B}})$ is the angle of the rotation which turned the orientation from $\underline{\underline{A}}$ to $\underline{\underline{B}}$.

In the search of the particle orientation, one last inconvenience of Euler angles is that they are not locally orthogonal, in the sense that

$$d(\{\theta_x, \theta_y, \theta_z\}, \{\theta_x + \Delta\theta_x, \theta_y + \Delta\theta_y, \theta_z + \Delta\theta_z\})^2$$

$$\approx \Delta\theta_x^2 + \Delta\theta_y^2 + \Delta\theta_z^2 + 2\Delta\theta_x \cdot \Delta\theta_z \cdot \sin(\theta_y) \qquad (6)$$

for Δ small. As a consequence, a uniform spacing of the Euler angles in $\theta_x, \theta_y, \theta_z$ does not sample the space of possible orientations in an optimal way. In particular near gimbal locks, the sampling rate would be higher at no higher accuracy. The so called Lattman angles[23]

$$\{\theta_+, \theta, \theta_-\} \equiv \{\theta_x + \theta_z, \theta_y, \theta_x - \theta_z\} \qquad (7)$$

fulfill local orthogonality since they verify

$$d(\{\theta_+, \theta, \theta_-\}, \{\theta_+ + \Delta\theta_+, \theta + \Delta\theta, \theta_- + \Delta\theta_-\})^2$$

$$\approx \frac{\Delta\theta_+^2(1 + \sin\theta)}{2} + \Delta\theta^2 + \frac{\Delta\theta_-^2(1 - \sin\theta)}{2}. \qquad (8)$$

As they are locally orthogonal, it is sufficient for sampling purposes to keep $\Delta\theta_+^2(1 + \sin\theta)/2$, $\Delta\theta^2$ and $\Delta\theta_-^2(1 - \sin\theta)/2$ constant. After a constant sampling of N values of θ with $\Delta_{\mathrm{Latt}} \equiv \Delta\theta = \pi/(N - 1)$, the stepping in θ_+ and θ_- can be computed with $\Delta\theta_+(\theta) = \Delta_{\mathrm{Latt}}/\sin(\theta/2 + \pi/2)$ and $\Delta\theta_-(\theta) = \Delta_{\mathrm{Latt}}/\sin(\pi/2 - \theta/2)$. It should be emphasized, that $\theta_- \in [0, 2\pi[$ whereas $\theta_+ \in [0, 4\pi[$. The Lattmann angles allow us to sample the set of orientations in an optimal way, in terms of achieving the best resolution from the point of view of the metric given by Eq. (8), and also from an algorithmic point of view.

Finally, in several instances it is convenient to describe a rotation by the direction of an axis $\hat{\boldsymbol{n}}$ about which the systems is rotated by an amount ϕ. The corresponding rotation matrix can be computed using the Rodrigues formula[20,22]

$$\underline{\underline{R}}(\hat{\boldsymbol{n}}, \phi)$$

$$= \begin{bmatrix} c\phi + n_x^2 A & -n_z s\phi + n_x n_y A & n_y s\phi + n_x n_z A \\ n_z s\phi + n_x n_y A & c\phi + n_y^2 A & -n_x s\phi + n_y n_z A \\ -n_y s\phi + n_x n_z A & n_x s\phi + n_y n_z A & c\phi + n_z^2 A \end{bmatrix}$$

with $A = (1 - c\phi)$. $\qquad (9)$

Equation (9) also allows us to extract the axis, $\hat{\boldsymbol{n}}$, and the angle, ϕ from any arbitrary rotation matrix. As a result, changing the coordinate system or changing the representation of rotation can be done by expressing the orientation in its matrix form, applying the transformation which changes the coordinate system and extracting the desired representation.

2. Angular velocity and acceleration

Angular velocity and acceleration are often obtained by direct differentiation of a time series of Euler angles, *e.g.*, using Eq. (3). However, it is possible to obtain the angular

velocity in the particle frame directly from the matrices. This technique is not sensitive to Gimbal locks because of the uniqueness of the orientation matrices.

Let $e_{x,y,z}^{\mathbb{P},k}$ be the particle coordinate system at time step k, whereas the fixed lab coordinate system is $e_{x,y,z}^{\mathbb{L}}$. For two time steps, k and $k+m$, we know the corresponding orientation matrices which rotate the particle

$$\underline{\underline{\mathbf{R}}}(\theta_k) \quad : \quad e_{x,y,z}^{\mathbb{L}} \quad \xrightarrow{\underline{\underline{\mathbf{R}}}(\theta_k)} \quad e_{x,y,z}^{\mathbb{P},k}$$

$$\underline{\underline{\mathbf{R}}}(\theta_{k+m}) \quad : \quad e_{x,y,z}^{\mathbb{L}} \quad \xrightarrow{\underline{\underline{\mathbf{R}}}(\theta_{k+m})} \quad e_{x,y,z}^{\mathbb{P},k+m}$$

$$\underline{\underline{\mathbf{T}}} \quad : \quad e_{x,y,z}^{\mathbb{P},k} \quad \xrightarrow{\underline{\underline{\mathbf{R}}}(\theta_{k+m})\underline{\underline{\mathbf{R}}}(\theta_k)^T} \quad e_{x,y,z}^{\mathbb{P},k+m}$$

in which the matrix $\underline{\underline{\mathbf{T}}}$ is the change in orientation, in other words the matrix representation of the discrete angular velocity (for a given time difference). The change is with respect to the particle coordinate system at time k: $e_{x,y,z}^{\mathbb{P},k}$. $\underline{\underline{\mathbf{T}}}$ expressed in the axis–angle convention [see Eq. (9)] returns a direction vector, $\hat{\boldsymbol{n}}$, of length unity and an angle, $\Delta\phi$ (meaning that between times k and $k+m$ the particles has rotated an angle $\Delta\phi$ around the vector $\hat{\boldsymbol{n}}$). The time difference, Δt, between the steps is a function of m. Therefore an estimator of angular velocity is

$$\boldsymbol{\omega}^{\mathbb{P}}(t(k)) = \frac{\Delta\phi}{\Delta t(m)}\left(n_x \cdot \boldsymbol{e}_x^{\mathbb{P},k} + n_y \cdot \boldsymbol{e}_y^{\mathbb{P},k} + n_z \cdot \boldsymbol{e}_z^{\mathbb{P},k}\right). \quad (10)$$

Averaging $\hat{\boldsymbol{n}}\frac{\Delta\phi}{\Delta t}$ over several separations, m, returns the angular velocity in the particle frame without a prior unwrapping nor problems near gimbal locks. The angular velocity with respect to the lab coordinate system is defined as

$$\boldsymbol{\omega}^{\mathbb{L}}(t(k)) = \underline{\underline{\mathbf{R}}}\left(\theta_k\right)^T \boldsymbol{\omega}^{\mathbb{P}}(t(k)). \quad (11)$$

The angular acceleration in either particle or lab frame is defined as

$$\boldsymbol{\alpha}^{\mathbb{L}/\mathbb{P}} = \frac{d}{dt}\boldsymbol{\omega}^{\mathbb{L}/\mathbb{P}}. \quad (12)$$

In practice, it is obtained from a convolution of the angular velocity time series with the derivative of a Gaussian kernel. This technique has proved to be efficient in removing noise.[4]

III. TRACKING

A. Position

Although the identification of a large sphere from the camera images causes no particular conceptual difficulty, the fact that the sphere is *textured* raises some practical issues. A simple thresholding returns only either the white or the black part of the particle. Reflections from the impellers continuously change the background, and small impurities in the flow and possible bubbles add sharp gradient noise to the images. Furthermore, the illumination of the flow is not perfectly uniform, and thus, shadows as well as reflections occur.

For each movie and for each camera, we compute the background view as the average of an equally distributed subset of its images. For each frame we then subtract the background and perform a *difference of Gaussians* blob detection. The threshold is adjusted by hand for each camera and light

arrangement. MATLAB's image-processing toolbox is used to identify blobs with a round shape and a diameter close to that of the particle. Shadows, bubbles, and reflections might be found during blob detection because of their sharp separation from the background, but they are of uniform texture and hence characterized by a small value of the variance of light intensity across the blob. The blob with highest variance and closest resemblance to a sphere is considered to be the particle. The precise position of the particle is refined using a *Canny edge* detection in a tight region around the blob. For each time step we record the position, (x, y), of the particle on the image in pixels plus its diameter, $2r$, and the deviation from the spherical shape as an error estimator. Since only one particle is placed into the flow, the track assembly is straightforward. The algorithm may temporarily loose the particle for short times (because of bad light reflection, blurs, ...); this is compensated by the large oversampling and gaps of less than five frames are interpolated to obtain longer tracks. Outliers are identified using a least square spline and replaced by an interpolation.

Tsai's camera model and calibration technique[24] is used to project the 2D positions into 3D. The calibration of the cameras contains the position of the camera plus its rotation with respect to the lab coordinate system, which is needed later for the orientation processing.

B. Orientation

The algorithm used to process the camera images and obtain a time series of orientations (and angular velocities) can be split into three parts: (i) by comparison of the sphere's picture with synthetic images, the algorithm identifies a set of possible orientations; (ii) from the set of possible candidates at successive instants, a *flow* algorithm identifies a likely time series; (iii) a post-treatment adjusts remaining ambiguities. These steps are described in details in this section.

1. Candidate finding

a. Synthetic images. The first step is to obtain a 2D projection, $\underline{\underline{\mathbf{S}}}(\underline{\theta})$, of a sphere with known texture and size at an arbitrary orientation, $\underline{\theta}$. This rendering is achieved using OPENGL, *via* the PSYCHTOOLBOX extensions for MATLAB—for a disk image of about 60 pixels, the algorithm can render several thousand orientations per second (see Fig. 3 for an illustration).

b. Texture extraction. Once the particle position and diameter are known, one extracts a disk subset of the image, centered on the particle, $\underline{\underline{\mathbf{G}}}$. In the first step the contrast is adjusted such that the global histogram of intensity contains at least b percent of black and w percent of white pixels (the algorithm only takes into account the disk/particle region in $\underline{\underline{\mathbf{G}}}$). The adjustable parameters b, w are fixed to $b = w \sim 30\%$ which is the minimum amount of black/white pixel in an arbitrary orientation. In a second step, the image is thresholded using Otsu's method[25] for the global histogram as well as for two moving regions. The thresholded image, $\underline{\mathbf{I}}$, is adjusted such
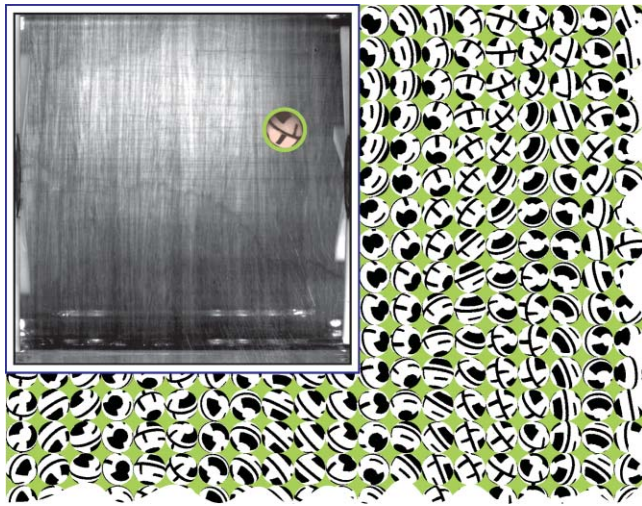
FIG. 3. (Color online) Synthetic 2D projections of the particle for a range of orientations, using OPENGL. A camera image of the moving particle is shown in the upper left corner (contrast enhanced; note the driving disks on either side).

that pixels outside the particle/disk are set to 0 whereas black is −1 and white +1. These steps are shown in Fig. 4.

c. Comparison, possible orientations. The image $\underline{\underline{\mathbf{I}}}$ (with diameter $2r$) obtained as above is ready for comparison with synthetic images. The resemblance to a rendered image $\underline{\underline{\mathbf{S}}}(\underline{\theta})$ with orientation $\underline{\theta}$ is estimated by

$$T(\underline{\underline{\mathbf{I}}}, \underline{\theta}) = \frac{1}{2} + \frac{1}{2\pi r^2} \sum_i \sum_j \underline{\underline{\mathbf{I}}}_{i,j} \cdot \underline{\underline{\mathbf{S}}}_{i,j}(\underline{\theta}), \qquad (13)$$

which is ratio of the number of correct pixels to the total number of pixels.

At this point we note that the computational cost of directly comparing an image $\underline{\underline{\mathbf{I}}}$ to synthetic ones $\underline{\underline{\mathbf{S}}}(\underline{\theta})$ covering the set of possible orientation $\{\underline{\theta}\}$ scales roughly as $\Delta_{\text{Latt}}^{-3}$, where Δ_{Latt} is the grid spacing in the orientation space. There is also the additional difficulty that the particle apparent diameter changes slightly as the sphere moves in the flows. For efficiency and physical correctness, we use the following strategy: instead of finding at any time step the best images, we identify a set of possible candidates for all time steps and then extract globally the time series of orientations.

First we render images, $\underline{\underline{\mathbf{S}}}(\{\underline{\theta}_{\text{coarse}}\})$, covering *all* possible orientations with a coarse grid—in practice $\Delta_{\text{Latt}} \approx 12°$.
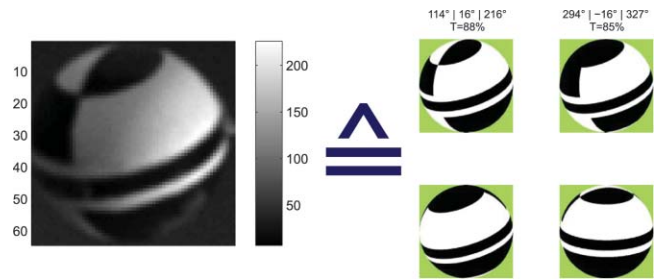

FIG. 5. (Color online) Particle camera image (left) and corresponding candidates, after analysis of the possible orientations (steps *1a–1c* described in the text).

Lattman angles are locally orthogonal and thus more efficient in creating such grids. The size of the rendered images is fixed to approximately one half of the particle's real diameter. Since their size does not change, these images are kept in the computer memory and do not need to be recomputed for every new image.

The thresholded particle image, $\underline{\underline{\mathbf{I}}}$, is then resized to the size of the renderings, $\underline{\underline{\mathbf{I}}}_{\text{coarse}}$, and compared to all synthetic images, $\underline{\underline{\mathbf{S}}}(\{\underline{\theta}_{\text{coarse}}\})$ as shown in Fig. 3 using Eq. (13). All angles $\underline{\theta}$ satisfying $T(\underline{\underline{\mathbf{I}}}_{\text{coarse}}, \underline{\theta}) > \max(T(\underline{\underline{\mathbf{I}}}_{\text{coarse}}, \{\underline{\theta}_{\text{coarse}}\})) - \delta_{\text{coarse}}$ are considered to be possible orientations. Here δ_{coarse} is an arbitrary thresholding value, with inspection showing that a value equal to 0.1 gives good results.

Experience shows that the identified possible orientations usually cover several broad classes. They are thus separated into groups of images whose orientations differ by less than a rough threshold, ~ 30–$45°$. For each group, synthetic images are further added using a fine grid spacing, $\Delta_{\text{fine}} = 3°$ (at this point "bad" images may cause the code to runaway; they are dropped and the code advances to the next time step). The possible orientations are then rendered in real size and compared [using Eq. (13)] to the image $\underline{\underline{\mathbf{I}}}$. For each group, the code returns the final best guess, i.e., the orientation with the maximum resemblance, thus drawing a list of *candidates*, see Fig. 5 for an example of a particle with its corresponding candidates.

2. Track assembly

After identifying the candidates for each time step, the most likely orientation for each time step has to be determined. However, the candidate with the highest count of correct pixels is not necessarily the best choice. Although
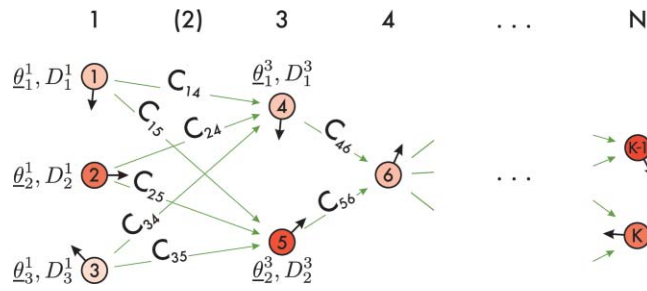

FIG. 4. (Color online) Texture extraction and comparison with a synthetic image. The resemblance between the image $\underline{\underline{\mathbf{I}}}$ and the synthetic projection $\underline{\underline{\mathbf{S}}}$ at angle $\underline{\theta}$ is estimated using Eq. (13).

FIG. 6. (Color online) Sketch of a graph connecting the possible candidates using the cost function C [cf. Eq. (15)]. For each time step t, we have $b(t)$ candidates with an orientation $\underline{\theta}_b^t$ and a resemblance D_b^t. In the time series $t = 1 \ldots N$, the candidates are labeled $k \in 1 \ldots K$. The directed graph C connects all candidates at t to all their next valid time step $t + \Delta t$; gaps are skipped as indicated for time step **2**.

counterintuitive, the direct use of two cameras seeing the particle at different angles does not simplify the problem, because in the case of a bad image, one camera falsifies the choice of the candidates found by the other camera. Moreover, gimbal locks prevent the use of a predictor–corrector scheme for the prediction of the orientation. However, the norm of angular velocity is assumed to be smooth and we search the time series which globally minimizes the sum $\sum_t \xi(t)$ along the time series of the so called direct neighbor distance function

$$\xi(t) \equiv \left| \boldsymbol{\omega}(\underline{\theta}(t), \underline{\theta}(t + \Delta t)) \right| = \frac{d(\underline{\theta}(t), \underline{\theta}(t + \Delta t))}{\Delta t} . \quad (14)$$

A direct neighbor is the next valid time step at $t + \Delta t$. The distance between two orientations does not depend on the representation, ensuring the robustness of the algorithm even at gimbal locks. Minimizing $\sum_t \xi(t)$ is only meaningful for small changes in orientation between two time steps, another requirement for high (over)sampling rates.

Flow algorithms are highly efficient in finding a global optimum for a discrete set of candidates. The following is done for each camera without considering the extra information from the second camera. In a first step we remove all candidates with a resemblance $T < s_{\text{quality}}$—in practice $s_{\text{quality}} = 0.5$. Then a directed graph is built which connects all candidates at time step t with all their direct neighbors at the nonempty time step $t + \Delta t$. The cost function is chosen such that it takes into account both the change in orientation and the quality of the matching

$$C(\{\underline{\theta}_A, T_A\}, \{\underline{\theta}_B, T_B\}) = d(\underline{\theta}_A, \underline{\theta}_B) \frac{2 - T_A - T_B}{\Delta t} \quad (15)$$

with $\{\underline{\theta}_A, T_A\}$ a candidate at time t and $\{\underline{\theta}_B, T_B\}$ a directly neighboring candidate at $t + \Delta t$.

A Dijkstra path finding algorithm returns the sequence of candidates having a global minimum of the total cost, i.e., the global minimum of change of orientation (weighted by the image quality) (cf. Fig. 6). In most cases this algorithm returns directly the time series of absolute orientation. Nevertheless, bad images introduce false candidates forcing the path finding algorithm to take a different, non-

physical path. These points manifest as spikes in the direct neighbor distance function, $\xi(t)$. After a spike, there is no guarantee that the path is still physical. Therefore, we segment the time series based on the spikes. The second view (from the second camera) treated with the same algorithm contains the information to correct such wrong segments. From the camera calibration the rotation matrix which transforms the orientations seen by one camera into the coordinate system of the other one, is known. Therefore, both views are expressed in an intermediate, common coordinate system where the segments with $d(\underline{\theta}_{\text{cam1}}, \underline{\theta}_{\text{cam2}}) \gtrsim 30°$ can be corrected.

The algorithm presented so far assumes an orthographic view. This condition holds only true if the particle center is on the optical axis of the camera or in the case one uses telecentric lenses. In the present experiment we do not, and the perspective effect alters the measured orientation (note that the parallax displacement corresponds to a change in the 2D projection, and hence to a rotation). The distortion induced by the perspective is characterized by the position of the particle center in the camera image, $\boldsymbol{X}$, and the focal length, f. Common camera objectives allow only small angles, $\gamma_{\text{persp}} \equiv \text{atan}(\|\boldsymbol{X}\|/f) \lesssim 15°$. As a consequence we assume that the shape of the particle does not change and we introduce an orientation matrix $R_{\text{persp.}}$ [taking advantage of the Rodrigues formula Eq. (9)]

$$\underline{\underline{R}}_{\text{persp.}}(\boldsymbol{X} = (x, y), f) = \underline{\underline{R}} \left(\frac{(-y, x, 0)}{\sqrt{x^2 + y^2}}, \text{atan}\left(\frac{\|\boldsymbol{X}\|}{f} \right) \right)$$

$$(16)$$

such that the measured orientation is related to the absolute orientation $\underline{\theta}_{\text{abs}}$ by $\underline{\underline{R}} \cong \underline{\underline{R}}_{\text{persp.}} \underline{\underline{R}}(\underline{\theta}_{\text{abs}})$. The perspective distortion can then be removed from the orientation time series.

Finally, after correcting for perspective distortion, a combined time series of orientation can be built using the information from both views, if they are expressed in the same coordinate system. Euler angles are not locally orthogonal, hence, we use the weighted mean of the orientation expressed in the axis–angle representation. The variance within a moving window of the direct neighbor distance function, $\xi(t)$, proves to be a good error estimator of the noise, since for short times the particle is assumed to rotate smoothly. A sample orientation track is shown in the upper panel of Fig. 7.

C. Robustness

A full study of the accuracy and robustness considering all possible distortions is beyond the scope of this article. In practice, the problems with real images are mainly caused by reflections, bad illumination, and objects (such as bubbles or dirt particles) between the particle and the camera. The setup, light conditions, and particle texture must be first tuned in order to optimize these parameters—by trial and error methods. For the orientation algorithm *per se*, we have used a series of synthetic images of known orientation. We found that the measurement error is $2°$, which is
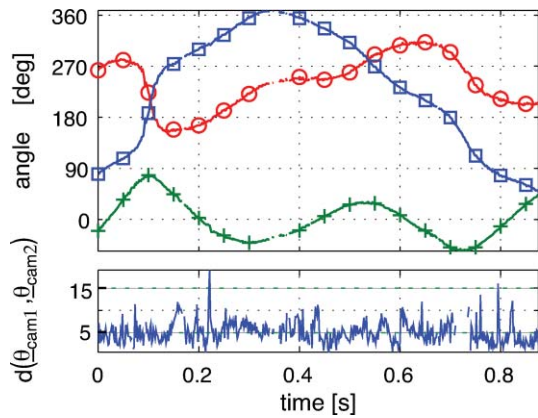
FIG. 7. (Color online) A sample orientation track; it is $\theta_x = \bigcirc$, $\theta_y = +$, $\theta_z = \square$, the bottom plot shows the distance (in degrees) between the independent orientation measurements from the two cameras.

smaller than the size, $\Delta_{fine} = 3°$ of the fine grid used in the image processing (cf. paragraph III B 1 c). A finer grid would improve the resolution for ideal images, but not for real images which, as stated above, always contain some amount of distortions or impurities. In addition, the fast dynamics of the particle and high frame rate ensure that wrong detection do not persist for longer than a few frames. As a result, most defects are detected and skipped or interpolated or handled as part of postprocessing (wrong orientations correspond to jumps in the direct neighbor distance function).

We illustrate the accuracy of the detection on two examples. The first one concerns the agreement between the orientation as estimated from each camera measurement. In the upper panel of Fig. 7, the combined three angles with respect to the lab coordinate system are plotted. The lower panel shows the distance (in degrees of angle) between the two estimations, $d\left(\underline{\theta}_{cam1}, \underline{\theta}_{cam2}\right)$. The probability density function (PDF) of these distances, computed with and without processing for perspective corrections are shown in Fig. 8. Correcting the systematic error induced by the perspective distortion reduces the mean value and width of the distribution. The remaining error is of random nature. Combining the

two independent views as described early leads to a weighted error of $\sim 3°$.

IV. RESULTS

The results in this section correspond to the flow created by counter-rotation of the driving disks at a rate of 3 Hz. In this case the net power injection is of the order of $\epsilon \sim 1.7$ W/kg, a value in agreement with bulk scalings and measurements, as reported in Ref. 26. The integral time scale T_L is about 0.3 s, so that the dissipative time and space scales are $\eta \sim 30\ \mu$m and $\tau_\eta \sim 1$ ms. As a result, the particle tracked has a size corresponding to $D/\eta \sim 600$ and $D/L_{int} \sim 0.6$ (L_{int} is the scale at which energy is fed into the flow). The flow Reynolds number based on the Taylor microscale is $R_\lambda \sim 300$. The camera frame-rate is 600 Hz, and the trajectories analyzed have been selected so that their duration is longer than $0.25 T_L$ and most range between 0.5 and $3 T_L$.

Figure 9 shows a histogram of the duration of recorded tracks for which the 6D coordinates of the particle are recorded. It has an exponential tail (as it was also the case when using acoustic tracking[27]). For very small times the histogram is biased by the fact that tracks shorter than 50 contiguous frames are discarded. Note also that long tracks are likely to correspond to trajectories spanning the flow volume, i.e., a spatial extend over which the large scale (anisotropic) circulation cannot be ignored.

However, one first result is that the particle explores uniformly the orientation space. This is seen in Fig. 10 showing the probability distribution functions of the Euler angles: as expected from a random distribution of orientations, the θ_x and θ_z components have a flat distributions spanning a $[-\pi, +\pi[$ interval, with the inner angle θ_y having a $cos(\theta_y)$ distribution over $[-\pi/2, \pi/2]$.

Interesting features are observed for the rotation dynamics. The statistics of angular velocity fluctuations are shown in Fig. 11. The distributions are symmetric. The three components, with respect to the lab coordinate system, follow the same statistics. This reflects the spherical symmetry of the particle; furthermore, it also shows that the turbulent swirls at the scale of the particle have no preferred orientation. The
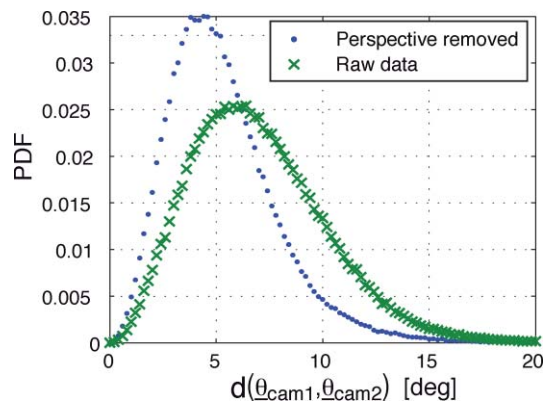

FIG. 8. (Color online) Probability density function (PDF) of the distance between the orientations measured from cameras 1 and 2, without correction for perspective distortion ($\times$) and with it ($\circ$).
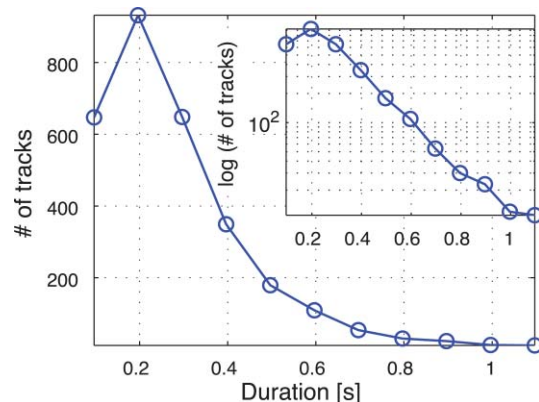

FIG. 9. (Color online) Histogram of track segments. The exponential decay rate is of the order of the integral time T_L of the flow.
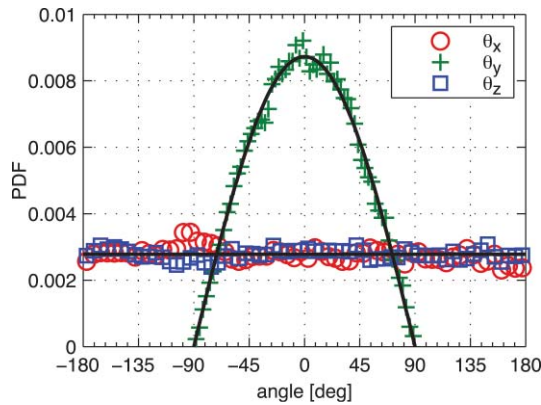
FIG. 10. (Color online) PDF of the orientation $\underline{\theta} = \{\theta_x, \theta_y, \theta_z\}$, the solid lines correspond to a uniform sampling of the orientation space.

mean of the angular velocity components (with respect to the lab reference frame) is essentially zero, up to statistical error. The *rms* amplitude of angular velocity fluctuation is of the order of 12 rad/s, see Table I, which is of the order of $u_{\rm rms}/D = 30$ rad/s. That is, it corresponds to the rotation that would result from imposing a velocity difference equal to almost $u_{\rm rms}$ across the diameter D of the sphere. Note that it is also of the order of the rotation rate of the driving disks. The PDF themselves displays weakly stretched-exponential tails; for a quantitative estimation we use the fitting function

$$\Pi_a(x) = \frac{e^{3a^2/2}}{4\sqrt{3}} \left(1 - \mathrm{erf} \left(\frac{\ln \left| x/\sqrt{3} \right| + 2a^2}{a\sqrt{2}} \right) \right), \quad (17)$$

which has been used extensively in the analysis of the intermittency of the translational motion of Lagrangian tracers[4] —it stems from the approximation that the norm of the vector has a lognormal distribution. For the angular velocity, one finds a fitting parameter $a = 0.45$, which corresponds to a flatness factor $F = 4$. It would be $F = 3$ for Gaussian statistics, so that our measurements show only a slightly non-Gaussian behavior for the angular velocity. This differs
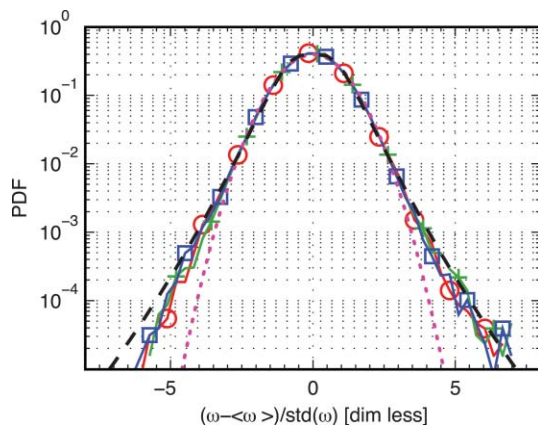
TABLE I. Characteristic values (mean $\pm$ rms) for the particle motion. The angular variables are given for the lab and particle coordinate systems.

		x	y	z	Norm
$\boldsymbol{v}$	(m/s)	0 ± 0.28	0 ± 0.40	0 ± 0.37	0.6 ± 0.2
$\boldsymbol{a}$	(m/s^2)	-0.2 ± 5.6	-0.3 ± 5.8	0.2 ± 6.3	8.5 ± 5.6
$\boldsymbol{\omega}^{\mathbb{L}}$	(rad/s)	0 ± 7.7	0.1 ± 7.3	0.1 ± 7.2	11.6 ± 5.5
$\boldsymbol{\omega}^{\mathbb{P}}$	(rad/s)	0 ± 6.5	-0.2 ± 7.8	-0.1 ± 7.8	
$\boldsymbol{\alpha}^{\mathbb{L}}$	(rad/s^2)	0 ± 540	0 ± 480	0 ± 420	670 ± 580
$\boldsymbol{\alpha}^{\mathbb{P}}$	(rad/s^2)	0 ± 510	0 ± 440	0 ± 460	

from the translational velocity, which is found to be slightly sub-Gaussian.

The angular acceleration has a strong non-Gaussian behavior, as seen in Fig. 12. Again, the three components follow identical statistics: there is no preferred direction for the torques acting on the moving sphere (with respect to the lab reference frame only—the issue of lift forces is addressed elsewhere[12]). The *rms* amplitude of angular acceleration is about 700 rad/s² , see Table I, again of the order of $(u_{\rm rms}/D)^2$. The statistics is strongly non-Gaussian, a fit using the same stretched exponential distribution yields $a = 0.6$, i.e., a flatness factor $F \sim 7.6$. The angular acceleration can be viewed as an angular velocity increment over a very short time lag. Hence, the PDFs of angular velocity increments change shape with the length of the time lag—from the one in Fig. 11 for small time increments to the one in Fig. 12 for integral times.

For comparison, we recall some features of the translational dynamics of the particle. It has statistical characteristics which are very close to the one reported for neutrally buoyant inertial particles with a size much closer to the dissipation scales of turbulence.[3,9–11] The translational velocity follows a Gaussian distribution, its acceleration is strongly non-Gaussian, with stretched exponential tails. Using the stretched exponential distribution leads to $a = 0.6$. One thus observes that the angular variables have intermittent dynamics, just as the translational motion. The fact that it is quite
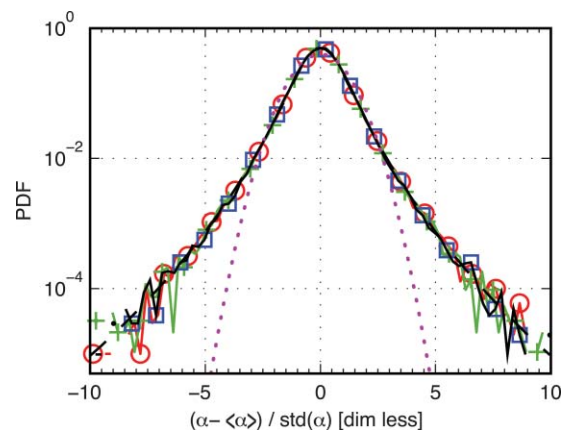

FIG. 11. (Color online) PDF of the (normalized) components of the angular velocity, $\omega_x = \bigcirc$, $\omega_y = +$, $\omega_z = \square$, the dotted curve is a Gaussian and the dashed one shows a stretched exponential with $a = 0.45 (F = 4)$.

FIG. 12. (Color online) PDF of angular acceleration; it is $\boldsymbol{\alpha}_x = \bigcirc$, $\boldsymbol{\alpha}_y = +$, $\boldsymbol{\alpha}_z = \square$, the dotted curve is a Gaussian and the dashed one shows a stretched exponential with $a = 0.6 (F = 7.6)$.

pronounced, even for an object of size close to the integral scale of motions came as a surprise and deserves further investigations.

V. CONCLUDING REMARKS

The focus of the work reported here has been to establish a technique for the study of angular and translational motion of a particle freely advected by a turbulent flow. We have shown that the measurement technique is robust, efficient, and accurate. As an application, we report here the first observation of intermittency for the rotational dynamics of an inertial particle. It should be pointed out, that the technique can be easily extended to track multiple spheres in a flow. Suitable algorithms for the tracking of multiple particle are presented in, e.g., Ref. 13. Smaller particle sizes are possible, if the particle texture is sufficiently well resolved and accurately painted.

We note that the algorithm used to compute the angular velocity can be applied to a set of particle attached to a rigid body which are tracked using standard particle tracking algorithms. If one records the positions in space of three or more points, $P_1 \ldots P_N$ at time t and $t + \Delta t$, their motion can be split up into a translation of their center of mass plus a rotation. Once the translation part is subtracted, the rotation, $\underline{\underline{\mathbf{R}}}_{\text{kabsch}}$, of the points $P_1 \ldots P_N$ around their center of mass can be computed efficiently using Kabsch's[28,29] algorithm. $\underline{\underline{\mathbf{R}}}_{\text{kabsch}}$ is then the matrix representation of the change in orientation, and the angular velocity, $\boldsymbol{\omega}^{\mathbb{P}}$, (in the particle reference frame) at time t can be extracted as done here. It should be pointed out that, one does not gain access to neither the angular velocity in the lab reference frame, $\boldsymbol{\omega}^{\mathbb{L}}$, nor to the absolute orientation, $\underline{\theta}$.

The strong intermittency in the particle's rotation may eventually be traced back to the complex interaction between the particle and its wake. One notes that this is inherently a finite size effect; for particles with very small diameters (compared to the Kolmogorov length) the translational and rotational dynamics are note coupled. For larger particles, as in our case, the influence of rotation on the motion of the particle is of interest, and will be the object of further analysis. One may also note that the influence of the inhomogeneity at large scale must be clarified. Further measurements in a more isotropic turbulent flow (such as the Lagrangian Exploration Module[30]) are underway.

[1] F. Toschi and E. Bodenschatz, Annu. Rev. Fluid Mech. **41**, 375 (2009).
[2] A. La Porta, G. A.Voth, A. M. Crawford, J. Alexander, and E. Bodenschatz, Nature (London) **409**, 1017 (2001).
[3] N. Mordant, P. Metz, O. Michel, and J. Pinton, Phys. Rev. Lett. **87**, 214501 (2001).
[4] N. Mordant, A. Crawford, and E. Bodenschatz, Physica D **193**, 245 (2004).
[5] T. R. Auton, J. Hunt, and M. Prud'homme, J. Fluid Mech. **197**, 241 (1988).
[6] P. Lovalenti and J. Brady, J. Fluid Mech. **256**, 561 (1993).
[7] E. Loth and A. Dorgan, Environ. Fluid Mech. **9**, 187 (2009).
[8] N. Qureshi, M. Bourgoin, C. Baudet, A. Cartellier, and Y. Gagne, Phys. Rev. Lett. **99**, 184502 (2007).
[9] N. Qureshi, U. Arrieta, C. Baudet, A. Cartellier, Y. Gagne, and M. Bourgoin, Eur. Phys. J. B **66**, 531 (2008).
[10] R. Brown, Z. Warhaft, and G. A. Voth, Phys. Rev. Lett. **103**, 194501 (2009).
[11] R. Volk, E. Calzavarini, E. Lévêque, and J. F. Pinton, J. Fluid Mech. **668**, 223 (2011).
[12] Robert Zimmermann, Yoann Gasteuil, Mickaël Bourgoin, Romain Volk, Alain Pumir, and Jean-François Pinton, arXiv:1012.4381v1, submitted.
[13] *Springer Handbook of Experimental Fluid Dynamics*, edited by C. Tropea, A. Yarin, and J. F. Foss (Springer-Verlag, Berlin, 2007).
[14] J. Ye and M. C. Roco, Phys. Fluids A **4**, 220 (1992).
[15] M. B. Frish and W. W. Webb, J. Fluid Mech. **107**, 173 (1981).
[16] N. Ouellette, H. Xu, M. Bourgoin, and E. Bodenschatz, New J. Phys. **8**, 102 (2006).
[17] R. Monchaux, F. Ravelet, B. Dubrulle, A. Chiffaudel, and F. Daviaud, Phys. Rev. Lett. **96**, 124502 (2006).
[18] D. H. Brainard, Spatial Vis. **10**, 433 (1997).
[19] D. G. Pelli, Spatial Vis. **10**, 437 (1997).
[20] H. Goldstein, C. Poole, and J. Safko, Classical Mechanics [3rd ed.], Addison Wesley, New-York, 2002.
[21] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation* (Wiley, New York, 2003).
[22] E. W. Weisstein (1999). MathWorld—A Wolfram Web Resource. http://mathworld.wolfram.com/
[23] E. Lattman, Acta Crystallogr. **28**, 1065 (1972).
[24] R. Tsai, IEEE J. Rob. Autom. **3**, 323 (1987).
[25] N. Otsu, IEEE Trans. Syst. Man Cybern. **9**, 62 (1979).
[26] G. Zocchi, P. Tabeling, J. Maurer, and H. Willaime, Phys. Rev. E **50**, 3693 (1994).
[27] N. Mordant, Ph.D. dissertation, École Normale Supérieure de Lyon, 2001.
[28] W. Kabsch, Acta Crystallogr. **32**, 922 (1976).
[29] W. Kabsch, Acta Crystallogr. **32**, 827 (1976).
[30] R. Zimmermann, H. Xu, Y. Gasteuil, M. Bourgoin, R. Volk, J.-F. Pinton, and E. Bodenschatz, Rev. Sci. Instrum. **81**, 055112 (2010).