

# Accurate detection and complete tracking of large populations of features in three dimensions

Yongxiang Gao and Maria L. Kilfoil

Department of Physics, McGill University, Montréal, Canada H3A 2T8

[kilfoil@physics.mcgill.ca](mailto:kilfoil@physics.mcgill.ca)

**Abstract:** Localization and tracking of colloidal particles in microscopy images generates the raw data necessary to understand both the dynamics and the mechanical properties of colloidal model systems. Yet, despite the obvious importance of analyzing particle movement in three dimensions (3D), accurate sub-pixel localization of the particles in 3D has received little attention so far. Tracking has been limited by the choice of whether to track all particles in a low-density system, or whether to neglect the most mobile fraction of particles in a dense system. Moreover, assertions are frequently made on the accuracies of methods for locating particles in colloid physics and in biology, and the field of particle locating and tracking can be well-served by quantitative comparison of relative performances.

We show that by iterating sub-pixel localization in three dimensions, the centers of particles can be more accurately located in three-dimensions (3D) than with all previous methods by at least half an order of magnitude. In addition, we show that implementing a multi-pass deflation approach, greater fidelity can be achieved in reconstruction of trajectories, once particle positions are known. In general, all future work *must* defend the accuracy of the particle tracks to be considered reliable. Specifically, other researchers *must* use the methods presented here (or an alternative whose accuracy can be substantiated) in order for the entire investigation to be considered legitimate, if the basis of the physical argument (in colloids, biology, or any other application) depends on quantitative accuracy of particle positions.

We compare our algorithms to other recent and related advances in location/tracking in colloids and in biology, and discuss the relative strengths and weaknesses of all the algorithms in various situations. We carry out performance tests directly comparing the accuracy of our and other 3D methods with simulated data for both location and tracking, and in providing relative performance data, we assess just how accurately software can locate particles. We discuss how our methods, now applied to colloids, could improve the location and tracking of features such as quantum dots in cells.

© 2009 Optical Society of America

**OCIS codes:** (100.2960) Image analysis; (180.2520) Fluorescence microscopy; (100.6890) Three-dimensional image processing; (180.1790) Confocal microscopy

---

## References and links

1. U. Gasser, E. R. Weeks, A. Schofield, P. N. Pusey, and D. A. Weitz, "Real-Space Imaging of Nucleation and Growth in Colloidal Crystallization," *Science* **292**, 258–262 (2001).

2. A. M. Alsayed, M. F. Islam, J. Zhang, P. J. Collins, and A. G. Yodh, "Premelting at defects within bulk colloidal crystals," *Science* **309**, 1207–1210 (2005).
3. V. W. A. de Villeneuve, R. P. A. Dullens, D. G. A. L. Aarts, E. Groeneveld, J. H. Scherff, W. K. Kegel, and H. N. W. Lekkerkerker, "Colloidal Hard-Sphere Crystal Growth Frustrated by Large Spherical Impurities," *Science* **309**, 1231–1233 (2005).
4. E. R. Weeks, J. C. Crocker, A. C. Levitt, A. Schofield, and D. A. Weitz, "Three-dimensional Direct Imaging of Structural Relaxation Near the Colloidal Glass Transition," *Science* **287**, 627–631 (2000).
5. J. C. Crocker and D. G. Grier, "Methods of Digital Video Microscopy for Colloidal Studies," *J. Colloid Interface Sci.* **179**, 298–310 (1996).
6. E. R. Weeks and D. A. Weitz, "Properties of cage rearrangements observed near the colloidal glass transition" *Phys. Rev. Lett.* **89**, 957041 (2002).
7. R. P. A. Dullens, D. G. A. L. Aarts, and W. W. Kegel, "Direct measurement of the free energy by optical microscopy," *Proc. Natl. Acad. Sci. U.S.A.* **103**, 529–531 (2006).
8. P. Schall, D. A. Weitz, and F. Spaepen, "Structural Rearrangements That Govern Flow in Colloidal Glasses," *Science* **318**, 1895–1899 (2007).
9. C. J. Dibble, M. Kogan, and M. J. Solomon, "Structure and dynamics of colloidal depletion gels: Coincidence of transitions and heterogeneity," *Phys. Rev. E* **74**, 041403 (2006).
10. D. Thomann, D. R. Rines, P. K. Sorger, and G. Danuser, "Automatic fluorescent tag detection in 3D with super-resolution: application to the analysis of chromosome movement," *J. Microsc.* **208**, 49–64 (2002).
11. O. M. Lozano and K. Otsuka, "Real-time Visual Tracker by Stream Processing," *J. Sign. Process. Syst.* DOI 10.1007/s11265-008-0250-2 (2008).
12. P. J. Lu, P. A. Sims, H. Oki, J. B. Macarthur, and D. A. Weitz, "Target-locking acquisition with real-time confocal (TARC) microscopy," *Opt. Express* **15**, 8702–8712 (2007).
13. K. Jaqaman, D. Loerke, M. Mettlen, H. Kuwata, S. Grinstein, S. L. Schmid, and G. Danuser, "Robust single-particle tracking in live-cell time-lapse sequences," *Nature Methods* **5**, 695–702 (2008).
14. A. Serge, N. Bertaux, H. Rigneault, and D. Marguet, "Dynamic multiple-target tracing to probe spatiotemporal cartography of cell membranes," *Nature Methods* **5**, 687–694 (2008).
15. P. J. Lu, E. Zaccarelli, F. Ciulla, A. B. Schofield, F. Sciortino, and D. A. Weitz, "Gelation of particles with short-range attraction," *Nature* **453**, 499–504 (2008).
16. M. J. Saxton, "Single-particle tracking: connecting the dots," *Nature Methods* **5**, 671–672 (2008).
17. <http://physics.nyu.edu/grierlab/software.html>
18. A. Rahman, "Correlations in the Motion of Atoms in Liquid Argon," *Phys. Rev.* **136**, A405–A411 (1964).
19. W. K. Kegel and A. van Blaaderen, "Direct Observation of Dynamical Heterogeneities in Colloidal Hard-Sphere Suspensions," *Science* **287**, 290–293 (2000).
20. N. B. Simeonova, R. P. A. Dullens, D. G. A. L. Aarts, V. W. A. de Villeneuve, H. N. W. Lekkerkerker, and W. K. Kegel, "Devitrification of colloidal glasses in real space," *Phys. Rev. E* **73**, 041401 (2006).
21. C. Donati, S. C. Glotzer, P. H. Poole, W. Kob, and S. J. Plimpton, "Spatial correlations of mobility and immobility in a glass-forming Lennard-Jones liquid," *Phys. Rev. E* **60**, 3107–3119 (1999).
22. Y. Gao and M. L. Kilfoil, "Direct Imaging of Dynamical Heterogeneities near the Colloid-Gel Transition," *Phys. Rev. Lett.* **99**, 078301 (2007).
23. J.-P. Hansen and I. McDonald, *Theory of Simple Liquids* (Academic Press, London, 1986), 2nd ed.
24. C. P. Royall, A. A. Louis, and H. Tanaka, "Measuring colloidal interactions with confocal microscopy," *J. Chem. Phys.* **127**, 044507 (2007).
25. G. Marty and O. Dauchot, "Subdiffusion and Cage Effect in a Sheared Granular Material," *Phys. Rev. Lett.* **94**, 015701 (2005).
26. S. S. Blackman and R. Popoli, "Design and Analysis of Modern Tracking Systems," (Artech House, Norwood, MA, 1999).
27. <http://plutarc.sourceforge.net/>
28. M. C. Jenkins and S. U. Egelhaaf, "Confocal microscopy of colloidal particles: Towards reliable, optimum coordinates," *Adv. Colloid Interface Sci.* **136**, 65–92 (2008).
29. A. S. Clarke and J. D. Wiley, "Numerical simulation of the dense random packing of a binary mixture of hard spheres: Amorphous metals," *Phys. Rev. B* **35**, 7350–7356 (1987).
30. S. Wilhelm, B. Grobler, M. Gluch, and H. Heinz, *Confocal laser scanning microscopy* (Microscopy from Carl Zeiss)
31. <http://www.physics.mcgill.ca/kilfoil/downloads.html>
32. K. Chen, A. Kromin, M. P. Ulmer, B. W. Wessels, and V. Backman, "Nanoparticle sizing with a resolution beyond the diffraction limit using UV light scattering spectroscopy," *Optics Communications* **228**, 1–7 (2003).
33. T. Savin and P. S. Doyle, "Static and Dynamic Errors in Particle Tracking Microrheology," *Biophys. J.* **88**, 623–638 (2005).
34. J. C. Crocker and B. D. Hoffman, "Multiple Particle Tracking and Two-Point Microrheology in Cells," Published in *Methods in Cell Biology* **83**, 141–178 (2007).

## 1. Introduction

Time-lapse confocal microscopy has provided fundamental insights into some of the long-standing questions in physics, such as the driving mechanism of nucleation [1] and melting [2, 3] of crystals, and the glass transition [4], allowing for quantitative tests of theoretical predictions. The availability of numerous fluorescently tagged colloids and the high precision in determining the central positions of particles in the plane from the raw digitized images enabled by the feature finding process [5] has made this technology more appealing and accessible. Quantitative studies at the single particle level, whether to measure dynamical or static properties, rely heavily on the accuracy of the coordinates extracted by the image analysis algorithms from the raw images.

In their pioneering work, Crocker *et al.* developed algorithms for automatically finding features in two-dimensional (2D) systems to one-tenth of a pixel resolution [5], conceived initially for circular-shaped features with uniform intensity in single image planes. The essential process involves, for each feature in the image, an initial estimate for the center of the intensity distribution; a correction for the center; and further refinement through a single step or iteration of steps to achieve sub-pixel resolution. Many applications to colloids use an incomplete three-dimensional (3D) extension to this technique for feature finding of spherical objects applied to stacks of images acquired by confocal microscopy [1, 2, 3, 6, 7, 8, 9]. These applications only reach sub-pixel accuracy in the image plane ( $x, y$ ), but not in the direction perpendicular to the plane ( $z$ ). The feature centers tend to be at an integer or half-integer number of pixels along the  $z$  axis – this is the basis for the assertion of half-pixel accuracy in  $z$  [4] – even though statistically they should appear with equal probability at any fractional pixel positions, that is, with positions uncorrelated with the underlying pixel grid. This *pixel biasing*, also called pixel locking, has been a severe limitation of 3D feature finding algorithms. Because of the lack of precision in 3D, the analysis in these applications has been restricted to only the 1D (with respect to either  $x$  or  $y$  in the image plane) versions of the quantities of interest. Other dynamical and static properties simply cannot be measured without significant errors, in the absence of careful position refinement: for example, low resolution will give rise to a high noise floor in measurements of particle mean squared displacements, a standard transport property measured to reveal the microscopic dynamics. Low spatial resolution also gives rise to broadening of the nearest-neighbor peak in the experimentally-determined radial distribution function, a quantity which describes how the density of surrounding matter varies as a function of the distance from any given point. One approach to avoid pixel biasing is to use model fitting [10]. However, this approach as currently implemented is prohibitively slow with a large number (i.e.  $> 10\,000$ ) of features per image. Recent advances in GPU computing, such as that described for tracking of faces in real-time in [11], may significantly accelerate this algorithm in the future, making the approach practical for large data sets. GPU porting could make the approach we describe in this article much faster, so that the user would not have to trade off speed or accuracy; with an order of magnitude increase in *both* accuracy and speed.

We show that pixel biasing is *not* an inherent limitation of dense colloidal samples. We demonstrate that we achieve subpixel accuracy in  $z$ . We show that in 3D, several iteration steps are needed for convergence to subpixel accuracy in  $x$  and  $y$  (monitored by the disappearance of pixel biasing), unlike the situation in 2D position location where the first iteration is typically sufficient. We show that our 3D subpixel localization allows us to observe an important dynamical effect at the colloidal gel transition that would have been completely masked by the unrefined position data.

In addition to the improvements carried out for 3D feature finding, we have developed a method for particle tracking of dense systems in the face of heterogeneous or intermittent dynamics. This method represents a vast improvement over the roughly 85% of particles that are

typically tracked reliably in colloid physics systems displaying heterogeneous dynamics. In these types of systems, the fastest and slowest tracked particles are segregated out for special analysis, and thus robust and complete tracking is crucial. Our multipass tracking method is complementary to our implemented 3D feature-finding for obtaining high quality data for dynamics in non-dilute or intermittently-dynamic systems in real space. We demonstrate nearly 100% tracking of the particles with perfect fidelity, including the most mobile component, in a dense system with dynamical heterogeneities. This performance is not possible with the standard approach.

In many fields, particularly in biology, the next generation of sophisticated particle localization or tracking methods is being developed to meet challenges that require greater performance [12, 13, 14]. A recent article in this journal describes an algorithm that is optimized for performance, forming part of a real-time particle-location image-processing protocol [12]. In this case, the position of each particle is determined with sub-pixel accuracy in the  $z$ -direction, which was used for a large, subsequent colloid study [15]. In biology, determining the central position of particles is also an important step for investigating dynamics, and methods in existence are mainly developed for 2D. Recently, Jaqaman *et al.* [13] and Serge *et al.* [14] independently developed 2D methods for particle localization in cells. We address how our work compares with and fits into the context provided by these other recent papers.

All of this points to the general issue of comparison: just how accurately can software locate particles? A direct comparison to assess performance of the various algorithms has never been carried out, and is precisely what is advocated in the recent literature, for example in the commentary piece of Saxton [16]. Here we make direct quantitative comparison of our and other algorithms for 3D particle localization. The result demonstrates that our fractional-shifting method produces a five-fold improvement in 3D accuracy, and a four-fold improvement in pixel resolution in the axial direction, over other methods in use.

## 2. Full sub-pixel feature finding in three dimensions

### 2.1. Three-dimensional “fracshift”

The kernel for the refinement step in 2D feature-finding algorithms that confers subpixel accuracy, called “fracshift” [5], is illustrated in Fig. 1. After an initial estimate of the feature center at pixels  $(m, n)$  by a local maximum operator, the shift of the center  $(\epsilon_x, \epsilon_y)$  for integrated intensity under the mask for the feature is calculated. Since the pixels in the mask are in registry with those in the image, we can simply apply the following Eq. (1) [5]

$$\begin{pmatrix} \epsilon_x \\ \epsilon_y \end{pmatrix} = \frac{1}{M} \sum_{i^2+j^2 \leq w^2} \begin{pmatrix} i \\ j \end{pmatrix} I(m+i, n+j), \quad (1)$$

and the center is updated to  $(m + \epsilon_x, n + \epsilon_y)$ . To do further refinement, the step of calculating the shift of the center of the intensity distribution is repeated, but now for the integrated intensity under the mask centered at  $(m + \epsilon_x, n + \epsilon_y)$ . Now each pixel in the mask is not in registry with the pixels of the underlying raw image, and its intensity is calculated by a linear interpolation of the pixels it involves in the raw image, based on their areal contribution:

$$\begin{aligned} I &= I_1 * (1 - |\epsilon_x|) * (1 - |\epsilon_y|) + I_2 * |\epsilon_x| * (1 - |\epsilon_y|) \\ &+ I_3 * (1 - |\epsilon_x|) * |\epsilon_y| + I_4 * |\epsilon_x| * |\epsilon_y| \end{aligned} \quad (2)$$

For 3D, the shift of the center of the intensity distribution for a feature initially at  $(m, n, p)$  is

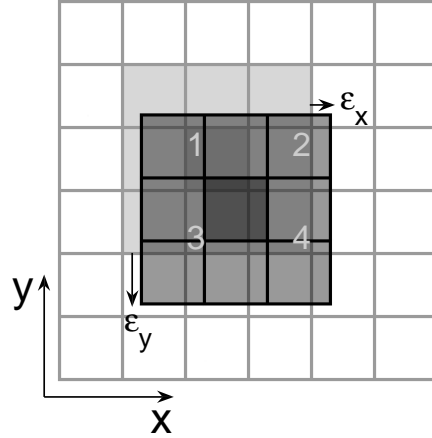


Fig. 1. Scheme for 2D fracshift. Gray lattices represent a digitized raw image. A mask centered at a local maximum  $(m, n)$  (which are integers) is shown as filled gray squares.  $(\epsilon_x, \epsilon_y)$  is the shift of the center of the intensity distribution from the local maximum, calculated based on Eq. (1) [5]. The mask for further refinement (fracshift), shown as filled black squares, is then centered at  $(m + \epsilon_x, n + \epsilon_y)$ . The central pixel of the mask is shaded in darker color. Now each pixel in the mask covers 4 pixels of the image. For example, the computed intensity under the center of the shifted mask involves pixels 1, 2, 3 and 4 in the original image.

calculated by analogy with the 2D case:

$$\begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \end{pmatrix} = \frac{1}{M} \sum_{i^2+j^2+k^2 \leq w^2} \begin{pmatrix} i \\ j \\ k \end{pmatrix} I(m+i, n+j, p+k) \quad (3)$$

where  $M$  is the integrated intensity and  $w$  is the radius of the mask. The feature center is then updated to  $(m + \epsilon_x, n + \epsilon_y, p + \epsilon_z)$ . This is where the 3D feature finding code widely used in real space imaging of colloids stops [17]. We do further refinement at this new position. In the 3D fracshift scheme, now each pixel in the new mask will involve eight pixels, four from each of two image planes sandwiching the mask. The intensity of each pixel in the new mask is:

$$I = \begin{cases} I_q * |\epsilon_z| + I_{q+1} * (1 - |\epsilon_z|), q = p \text{ if } \epsilon_z \geq 0 \\ I_q * (1 - |\epsilon_z|) + I_{q+1} * |\epsilon_z|, q = p - 1 \text{ if } \epsilon_z < 0 \end{cases} \quad (4)$$

Here  $I_q$  and  $I_{q+1}$  are the intensity of the pixels projected on the two raw image planes labeled  $q$  and  $q + 1$ , and their intensity is calculated using Eq. (4).

We check pixel biasing by examining the fractional part of the feature positions, which should be uniformly distributed between 0 and 1 for an unbiased case. In Fig. 2, we show the progression of this distribution with the number of iterations of refinement for  $x$  and  $z$ . Approximately

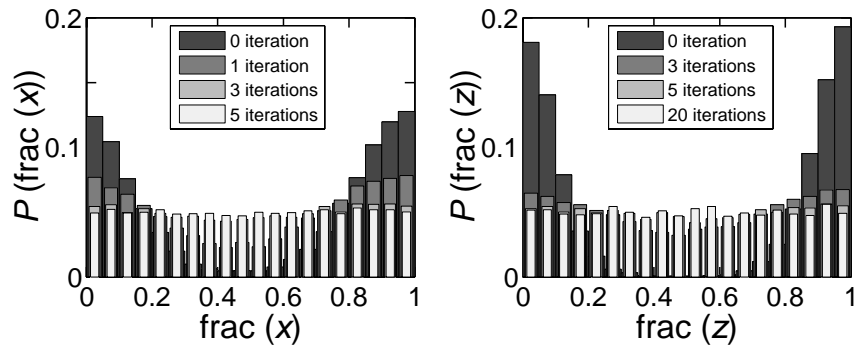


Fig. 2. Histograms of the fractional part of  $x$  (a) and  $z$  (b) with different numbers of iterations, derived from 17446 features. The distribution becomes stable and flat after approximately 5 iterations for  $x$ , and after approximately 20 iterations for  $z$ .

20 iterations are required to stabilize the center of the intensity distribution and achieve sub-pixel resolution. Pixel biasing in  $x$  and  $y$  is usually less severe than in  $z$ , due to the typically higher sampling in  $x$  and  $y$  compared to  $z$ . As demonstrated in Fig. 2, even  $x$  and  $y$  data suffer from pixel biasing from the 3D feature finding algorithms because dimension  $z$  is involved in the positioning of the mask during the refinement.

Important parameters for feature finding are the minimum separation allowed between two local maxima and the size of the mask used in the position refinement. These two parameters are often coupled. In a dense system, it is preferable to set them independently, allowing for tailoring of the mask to a size optimized for accurate finding of features that are touching. In this case we usually set the minimum separation to be approximately the radius of the features to find all the features. As a result there may be cases wherein two local maxima are found for the same particle. After a constant correction of the feature center by iterations of fracshift, such overlaps become trivial since their positions will converge.

We quantify the degree of refinement obtained with successive iterations of the position correction by using a  $\chi^2$ -test for uniform distribution of the fractional part of  $x$ ,  $y$  and  $z$ . In Fig. 3(a), we plot the degree of deviation from uniformly distributed particle positions against iteration number for  $P(\text{frac}(x))$  and  $P(\text{frac}(z))$  shown in Fig. 2.

## 2.2. Detailed shape of the mask used to find features

We show in Fig. 4(a) the linear interpolation scheme for the mask as a whole. We illustrate this for  $z$  as the mask size is typically small in  $z$ . The image layers are represented by black solid lines, the interpolated layers as dashed lines, the mask as blue (gray) solid lines, and the feature center initially identified as the red spot. We move the mask to this spot, integrate the intensity under the mask, and calculate the shift of the center of mass. Since the spot is located between two image layers, the algorithm performs a linear interpolation between neighboring layers, so that the red spot is located on one of the interpolated layers (i.e. in the shifted mask). The process is repeated for successive refinements of position.

The detailed shape of the mask matters a great deal for obtaining sub-pixel resolution. The mask should correspond as closely as possible to the feature. For dense systems, the mask should be slightly smaller than the feature, and for reasons of practicality, the mask should be sized an odd integer number of pixels. Since the sampling interval in  $z$  (the optic axis) is differ-

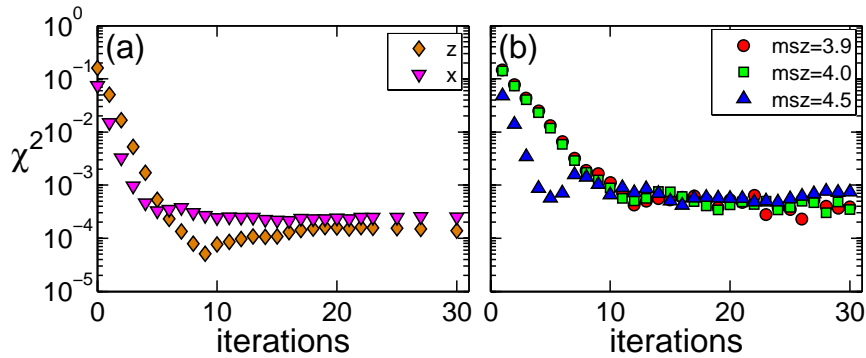


Fig. 3.  $\chi^2$ -test for uniform distribution of the fractional part of the refined pixel positions, plotted logarithmically to enhance the data where the deviation from flat distribution is so near zero. (a): for fractional distributions shown in Fig. 2 derived from 17446 features in the imaging volume, the  $\chi^2$  is shown for  $z$  and  $x$ . Mask size used is 15 in  $x, y$  and 3.9 in  $z$ . (b):  $\chi^2$  plotted for  $z$  from 1669 particles in a more dense sample, using mask size in  $z$  of 3.9, 4.0 and 4.5 .

ent than that in  $(x, y)$ , a mask for a spherical particle in real space is elliptical in voxel space. In fact, the  $z$ -sampling interval is intrinsically different because the optical transfer function blurs things in  $z$  more than in  $x, y$  with confocal microscopy. Although the mask shape in  $z$  is more strongly affected than in  $(x, y)$  by the exact choice of mask size for the first few refinement iterations because of the lower sampling in  $z$  than in  $(x, y)$  in our case, importantly, the algorithm is insensitive to these subtle mask shape changes: convergence to uniform distribution of  $\text{frac}(z)$  is always achieved following  $\sim 10$  refinement iterations, as shown in Fig. 3(b).

We note that this entire scheme works so flawlessly because the initial localization locates the features correctly to within a half of a pixel, so that cumulative successive shifts in position in any of  $x, y$  and  $z$  under all iterations is less than a pixel. This is the case when good initial parameters are used in the feature finding for the mask generation, which highlights the importance for following the above strategy for determining mask size.

### 2.3. Impact of sub-pixel resolution on measurements of dynamic and static quantities

As a demonstration of the utility of the extension to three dimensions for bulk colloidal studies, we first examine its effect on a dynamical quantity used to characterize the system behavior: the self part of the van Hove correlation function  $G_s(z, \tau) = 1/N \sum_i^N \delta[z + z_i(0) - z_i(\tau)]$ . The function  $G_s(z, \tau)$  measures the probability distribution of particle displacements at lag time  $\tau$  [18], which takes the form of a Gaussian function for particles with Fickian behavior, and otherwise a non-Gaussian but nevertheless smooth function, for example for a system close to the glass or jamming transition. In Fig. 5, we show that the curve of  $G_s(z, \tau)$  is smooth for the sub-pixel resolution data, while for data without refinement and thus suffering from strong pixel biasing,  $G_s(z, \tau)$  is non-monotonic and shows strong oscillations peaked at integer numbers of pixels in  $z$ . To our knowledge, experimental data for  $G_s$  has been shown only for  $x$  and  $y$  [4, 9, 19, 20] until now but not for  $z$ , due to the severe pixel biasing highlighted here that we have resolved.

We further demonstrate the impact of pixel biasing on experimental results by comparing the full three-dimensional particle mean squared displacements  $\langle \Delta r^2(\tau) \rangle$  (MSD), where the

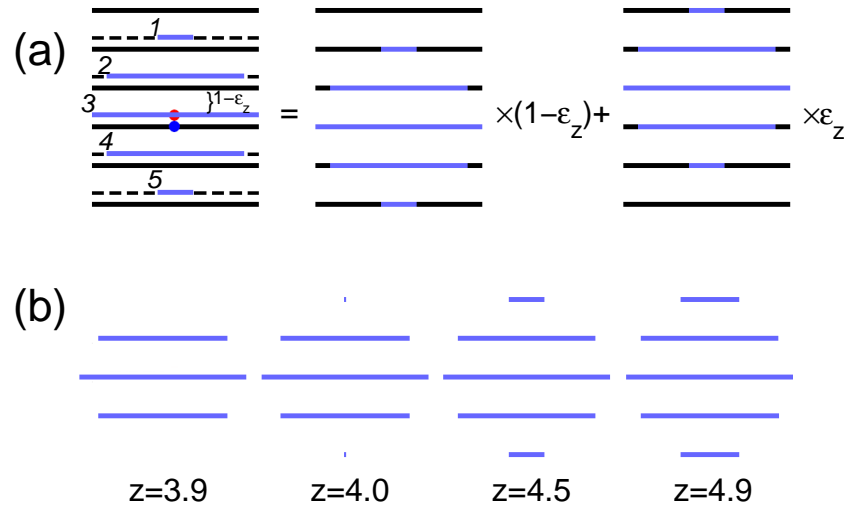


Fig. 4. (a) The fracshift linear interpolation scheme for the mask as a whole, illustrated for the  $z$  direction. The feature center initially identified is indicated by the red spot. The image layers are represented by black solid lines, the interpolated layers as dashed lines, and the mask as blue (gray) solid lines. (b) Non-integer values for the mask size in  $z$  are converted into minor changes in the shape of the mask. While this can allow for faster position refinement in the first few iterations of fracshift, successive refinements always result in convergence to uniform distribution of  $\text{frac}(z)$ , independent of these subtle changes of the mask shape (shown in Fig. 3 at right).

angle brackets indicate the autocorrelated motion averaged over all particles. For a system with Fickian behavior, the slope of the MSD in a logarithmic plot is 1, while for a system close to the glass or jamming transition, we expect a lag-time independent plateau in the MSD [6, 21] due to the caging effect. We observe in Fig. 6 that at short times, the data without refinement has a much higher noise floor due to the poor resolution. The MSD thus obtained tends to show a plateau, which can, in the worst case, cause experimentalists to conclude that the particles are glassy or in a solid-like medium, rather than that the apparent plateau is the limit of the experimental resolution.

We compare the 3D MSD of refined features to that of features without refinement in Fig. 6. The comparison is made on exactly the same set of slow particles identified in [22] in a colloidal suspension at a gel transition. The error due to feature finding is striking. At short times, the data without refinement has a much higher noise floor due to the poor resolution. We observe an actual plateau a whole 5 times lower in  $\langle \Delta r^2(\tau) \rangle$  using refined 3D position data. These results demonstrate that measurement of these quantities from 3D data requires accurate sub-pixel measurements.

We have also investigated the results for  $\langle \Delta x^2(\tau) \rangle$ . In Fig. 6 we show that if one plots simply the MSD for  $x$  or  $y$ , the results are still not reliable. The actual plateau in  $\langle \Delta x^2(\tau) \rangle$  obtained using the refined positions is a factor of  $\sim 6$  times lower in than the apparent plateau from



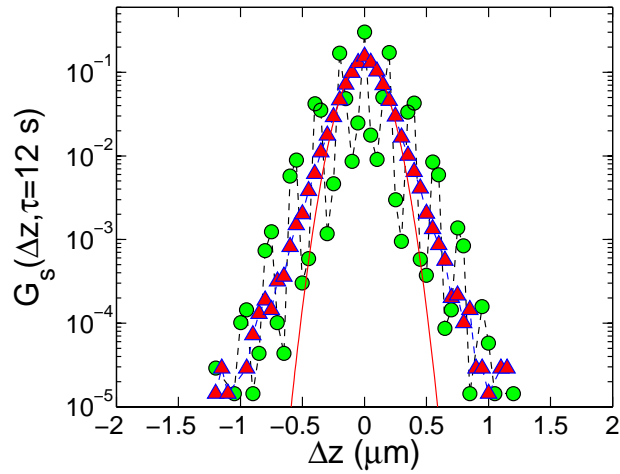


Fig. 5. The self van Hove correlation function for  $z$  at the shortest  $\tau$  of the experiment. The data was acquired at  $z$  spacing of  $0.2\mu\text{m}$  in  $z$ . Circles: no fracshift is employed. Triangles: 20 iterations of fracshift are employed. Dashed lines are guides to the eye. The solid line is a Gaussian lineshape fit to the data in which subpixel resolution in  $z$  has been obtained.

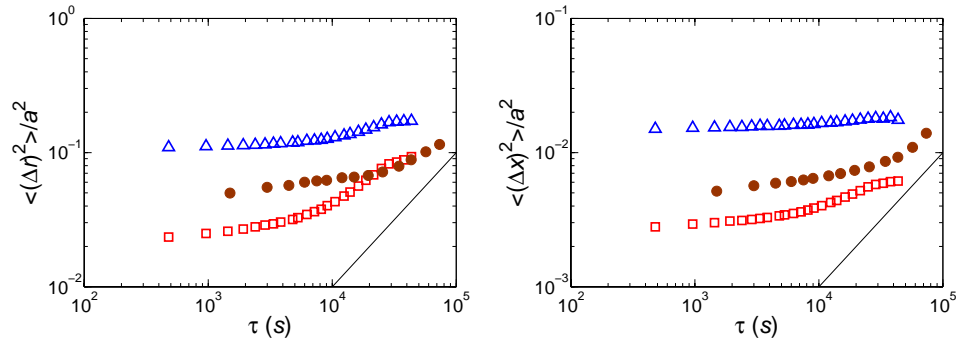


Fig. 6.  $\langle \Delta r^2(\tau) \rangle$  (left) and  $\langle \Delta x^2(\tau) \rangle$  (right) for a suspension of colloids with added attractive interaction at volume fractions close to the gel transition. Features without 3D subpixel refinement (blue triangles) and features with refinement (red squares), for a sample with short range attractive interaction of strength  $-3.02k_B T$  and colloid volume fraction  $\phi = 0.492$ . brown: features with refinement for moderate-to-long range attraction at interaction strength  $-2.86k_B T$  and at  $\phi = 0.440$ .

the unrefined data. The error in feature finding is large enough to mask useful information. This is worse still if one tries to extract a localization length scale from the MSD for caging or bonding length. To highlight this, we show in addition the MSD obtained from a second colloidal suspension at a longer range of attraction at the same interaction strength. With the position refined data we observe a different height of the plateau with two different ranges of attraction, an important effect that would have been completely masked by the unrefined data.

Pixel biasing also affects static properties. One typical function used to quantify the structure of the hard-sphere fluid at different densities is the (3D) radial distribution function  $g(r) = \rho^{-2} \langle \sum_i \sum_{j \neq i} \delta(\vec{r}_i) \delta(\vec{r}_j - \vec{r}) \rangle$ , which measures the deviation of the distribution of inter-

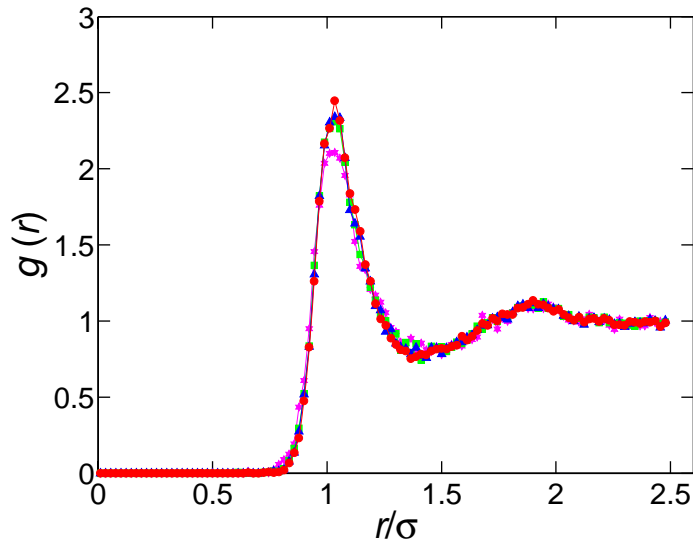


Fig. 7. The radial distribution function  $g(r)$  of the colloidal gel acquired at  $0.2\mu\text{m}$  spacing in  $z$  and analyzed with 0 (stars), 3 (squares), 5 (triangles) and 20 (circles) iterations of the position refinement.

particle separation from that of an ideal gas. The indices  $i$  and  $j$  run over all particles. As shown in Fig. 7, poor resolution in particle positions tends to lower and broaden the peak of  $g(r)$ , which weakens the reliability of measuring both thermodynamic quantities (for example, pressure, energy and compressibility) [23] and microscopic quantities (such as inter-particle potential [24]) from the height of the first peak of  $g(r)$ .

In conclusion, inaccurate sub-pixel resolution adversely affects measurements of static and dynamic quantities. Caution is therefore advised when using results derived from positional data without proper refinement in  $z$ .

### 3. 3D multipass tracking

In experiments, resolving the dynamics of individual particles is a difficult task for molecular liquids. This becomes attainable in the colloidal, biological, granular, worlds, where direct visualization is possible in time series of image (or image stack) frames [1, 2, 5, 7, 19, 25].

Tracking algorithms link the positions found in successive frames into trajectories (for a review, see [26]). The base algorithm we use is in Crocker *et al.* [5]. The features are linked into trajectories by minimizing the total displacement of individual beads between successive frames. The algorithm can be made to allow beads to skip one or more frames, with the trajectory continuing when they reappear. The search is limited to a maximum likely displacement between successive frames. Resolving single particle dynamics is straightforward where the particles are present in dilute quantities, for example in a dilute colloidal suspension or when used as probe particles. Sparse systems are characterized by  $d \sim v \times t$  and dense systems by  $d \ll v \times t$ , where  $d$  is the nearest neighbor distance,  $v$  is the particle velocity, and  $t$  is the time between frames.

For a dense system, tracking with 100 percent fidelity presents a much greater challenge when there is a broad distribution of mobilities inside the system, as is the case for a system close to the glass or jamming transition [4, 21, 25]. In Fig. 8(a), we show a simple homogenous

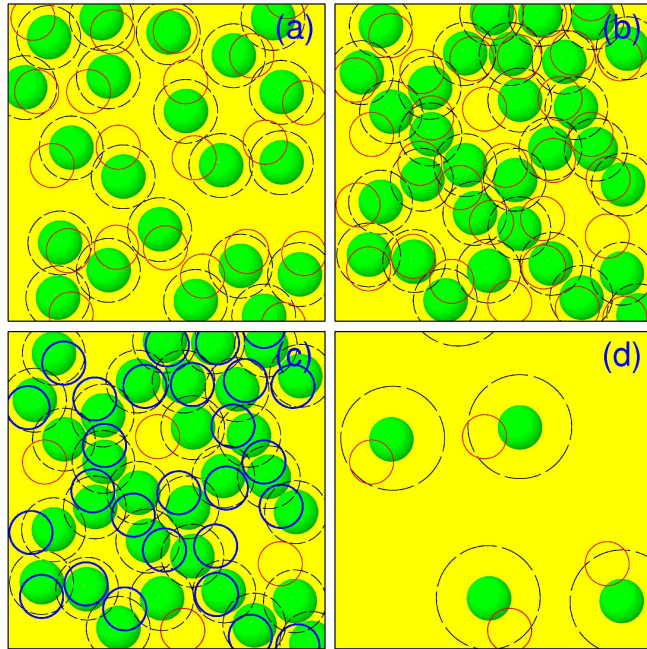


Fig. 8. (a, top left) Simple homogenous scenario for particle tracking. Green spheres: particle positions at time  $t_j$ ; red circles: particle positions at  $t_{j+1}$ ; black circles: range for searching particles belong to the same trajectories. (b, top right) Complex scenario with dynamical heterogeneities. (c, bottom left) Same as (b), with blue circles representing those particles in track. (d, bottom right) We remove of those particles that are in track and enlarge the searching range.

scenario, where each particle has only a small displacement during the time interval. Within one judiciously chosen cutoff distance, there is only one candidate for each particle. All the particles may be tracked easily with one pass. However, the scenario can be more complicated, as shown in Fig. 8(b), where dramatic dynamical heterogeneities are present. There is no one simple cutoff that one can use to track most of the particles reliably. Some of the tracks are lost if one uses a smaller cutoff. A larger cutoff can give rise to unreliable tracks. We solve the problem by employing a small cutoff first so that we can keep those particles having small displacements in track. Depicted in Fig. 8(c), particles with large displacement are not tracked in the first pass. We then remove from the position dataset those particles that have already been tracked, and increase the cutoff distance as depicted in Fig. 8(d) to track those particles with large displacements. These combined steps allow tracking of all particles with complete fidelity.

In Fig. 9, we show a typical case where by using a single cutoff, we are able to track only 86.5% of all the experimentally-determined particles with optimum choice of parameters  $r_c = 0.7 \mu\text{m}$ ,  $good = 20$  and  $mem = 3$ , where  $r_c$  is the maximum distance one searches for a candidate,  $good$  is the minimum number of times a particle must appear in the track, and  $mem$  is the maximum number times a particle can be missing before it reappears. With  $mem = 1$ , a much more preferred value for this parameter since we track the particles in 3D and do not expect a particle disappear for more than one frame, we have far fewer particles in track. To

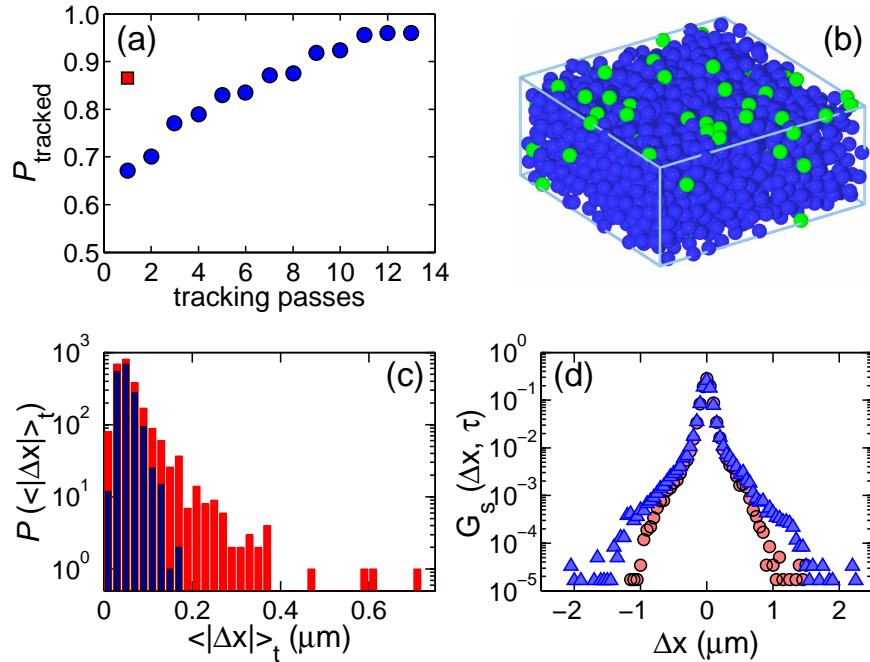


Fig. 9. (a) Total percentage of particles tracked by the end of each pass plotted against pass number. Square symbol: single pass tracking with optimized parameters  $r_c = 0.7\mu\text{m}$ ,  $good = 20$  and  $mem = 3$ . Circles: 13-pass strategy with  $mem = 1$ . Pass 1:  $r_c = 0.95\mu\text{m}$ ,  $good = 49$ ; 2:  $r_c = 1.20\mu\text{m}$ ,  $good = 49$ ; 3:  $r_c = 0.95\mu\text{m}$ ,  $good = 40$ ; 4:  $r_c = 1.25\mu\text{m}$ ,  $good = 40$ ; 5:  $r_c = 0.95\mu\text{m}$ ,  $good = 30$ ; 6:  $r_c = 1.25\mu\text{m}$ ,  $good = 30$ ; 7:  $r_c = 0.95\mu\text{m}$ ,  $good = 20$ ; 8:  $r_c = 1.25\mu\text{m}$ ,  $good = 20$ ; 9:  $r_c = 0.95\mu\text{m}$ ,  $good = 10$ ; 10:  $r_c = 1.25\mu\text{m}$ ,  $good = 10$ ; 11:  $r_c = 0.95\mu\text{m}$ ,  $good = 5$ ; 12:  $r_c = 1.25\mu\text{m}$ ,  $good = 5$ ; 13:  $r_c = 1.35\mu\text{m}$ ,  $good = 5$ . (b) Snapshot for the configuration after all these passes using the multipass strategy. Blue spheres are particles that are in tracks and in green are those not tracked. (c) Speed (averaged step size) histogram of the tracks obtained with multipass tracking (red), and with the standard approach (blue). Multipass tracking can capture the many large steps that are missed by the standard tracking. (d) Comparison of the self van Hove correlation function derived from the particle trajectories obtained via the two tracking methods, for a dense colloidal suspension at lag time  $\tau = 9600\text{s}$ . Triangles indicate the particles tracked via multipass tracking, and circles indicate the same particles tracked via the standard method. The anomalously large steps are clearly missed by the standard approach and are detected with the multipass tracking.

overcome this completely, we have developed a multipass strategy for particle tracking based on the above idea. We first impose a severe criterion by using smaller  $r_c = 0.95\mu\text{m}$  and a strict  $good = 49$  for the 49 time steps, to keep tracks that persist for the entire 50 stacks and have the most sluggish dynamics. We then separate these tracks from the particle list to be tracked, and relax the tracking parameters for the list of particles remaining. We increase the cutoff distance to  $r_c = 1.20\mu\text{m}$  and keep  $good = 49$ . This strategy is employed in successive passes with the priority given to complete tracks rather than to breaking the long tracks into shorter ones. Longer tracks will allow for statistical measurements over longer time scales. We then lower both the cutoff distance and  $good$  parameter to keep the second longest tracks with the strict mobility criterion, and remove them from the particle list to be tracked; and again in successive passes increase the cutoff distance to retain as many second-longest tracks as possible.

We repeat this low-cutoff, high-cutoff iterative process with gradual decreasing of the *good* parameter until the tracks are as short as 5 frames in length. “Birth” (appearance) and “death” (disappearance) of particles occurs at the faces of the imaging volume, and by restricting the multipass tracking to longer tracks first, we only track them when all the longer-lived particles are removed from the data set and eliminated as possible matches. Since large cutoff can give rise to incorrect identifications, we eliminate potential errors before we enlarge the cutoff distance. The total percentage of particles tracked by the end of each pass is plotted against pass number in Fig. 9(a). After all these passes, we have at least 96% particles in track for each experiment. All of the remaining untracked features are at the edge of the field of view, as shown in Fig. 9(b). For other experiments involving more time steps (140), we track greater than 98% of all the particles. Our multipass tracking strategy thus works with near-perfect fidelity even in the face of particle birth and death, where the number of features in each time step may vary.

The major advance of this technique is in the ability to track those particles with extreme displacements, keeping the tracks long, while retaining faithful tracking of the rest of the dynamics where steps are short and neighbors are plentiful. A mobility histogram of the tracks obtained as discussed above via the standard approach, and with our multipass tracking, is plotted in Fig. 9(c). The speed or mobility is defined for each particle based on its average magnitude of displacement during each time step  $\delta t$ ,  $\langle |\vec{r}(t + \delta t) - \vec{r}(t)| \rangle_{\delta t}$ , over the time series. The probability of distribution of the mobilities is plotted logarithmically to emphasize the small number of very highly mobile particles. The histogram demonstrates that with our method, we detect the fast fraction, whereas the standard approach fails to notice it. To demonstrate the impact of these missed displacements on the physical content carried by the particle dynamics, we plot in Fig. 9(d) the self part of the van Hove correlation function for particles tracked via the two methods. The self van Hove correlation function is the distribution of particle total displacements during some lag time  $\tau$  (here 9600s). These results demonstrate that as a result of the missing rare anomalously large steps in the standard approach, the breadth of the distribution is greatly underestimated. The multipass tracking is sensitive to both small vibrational motions as well as the larger displacements of the faster particles.

Finally, we demonstrate the fidelity of the tracking directly against particles with known identities for all time points by running the multipass tracking algorithm on simulated data. Here periodic boundary dictate that there is no birth or death of particles. 100% fidelity of tracks should be attainable in this case with our method. The simulation is carried out for 1000 particles at a dense volume fraction,  $\phi = 0.60$ , with glass-like dynamics. One configuration is visualized in Fig. 10.

The result of this test is that with multipass tracking, all of the particles are tracked with 100% fidelity over all 49 time steps. It was found to be impossible with conventional single-pass tracking to track all the particles with 100% fidelity at this same time step, no matter what the choice of cutoff distance. Running tracking on the same set of data without periodic boundary conditions and thereby allowing for particle birth and death, yielded 99% completeness with perfect fidelity for the multipass tracking, while the best performance that could be achieved with the standard approach at 93% completeness. The latter performance required very short cutoff distance such that many long tracks are sacrificed for shorter tracks, an underwhelming outcome if the quality of data relies on retaining the tracks over the long time series to obtain good particle statistics for long time metrics. An interesting future test of the multipass tracking would be to see what kind of velocity distributions force the algorithm to break down (if any).

#### 4. Comparison with other methods

Other feature finding algorithms are available for 3D particle localization in colloidal systems at subpixel resolution. Lu *et al.* adapted the Crocker and Grier 3D localization algorithm and

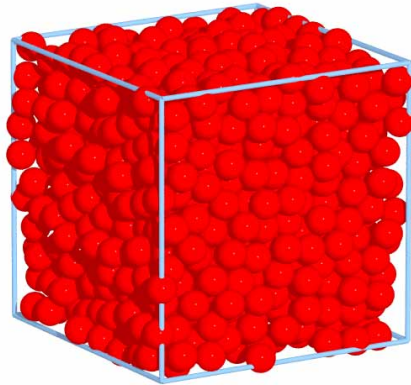


Fig. 10. A representative configuration of the simulation used to demonstrate the complete fidelity of the multipass tracking strategy.  $\phi = 0.60$ , with 1000 particles in the box.

optimized it for speed [12]. The Lu *et al.* code, available as open source on sourceforge.net [27], decomposes the 3D feature finding problem into a 2D one, which greatly reduces the processing time so that the motion of a target object can be followed in real time. The algorithm first identifies the in-plane  $x, y$  coordinates for a particle in each layer in which it appears, and uses the average in-plane coordinates as the final result for  $(x, y)$  centroid position. It then identifies the location of a particle in  $z$  as the peak position of a parabola fit along  $z$  to the intensity distribution integrated over the spot in each layer. Although their method involves a fast identification of the center of mass position of the largest object in the sample and moves the microscope stage to bring that point to the center of the 3D imaging volume, importantly, the Lu *et al.* code does not resolve the motions of individual particles over time. Jenkins *et al.* also developed a method to extract the coordinates of particles in 3D [28]. Part of their method is identical to that of the Crocker and Grier algorithm: the raw images are first filtered and then the local intensity maxima are identified. These authors then fit the intensity distribution of each particle in the raw images to an experimentally determined function which is obtained by averaging the intensity distribution of many particles. They then calculate the least square deviation of the real distribution to the fit for each particle. The rough center position is identified to be at the pixel that gives the minimum least square. They obtain subpixel resolution by doing a parabola interpolation for the least square in each direction:  $x$ ,  $y$  and  $z$ . The center is identified to be where the interpolated least square exhibits a minimum. They compare the radial distribution functions ( $g(r)$ ) for particle coordinates obtained by their algorithm and by the Crocker and Grier 3D algorithm. The peak of  $g(r)$  is sharper by their algorithm, indicating a better resolution. However, the error they estimate for their algorithm, 180 nm for  $x$  and  $y$ , and 300 nm for  $z$ , is too large for this method to be suitable over other available methods, and greater resolution is only achieved by successively rejecting particles that contribute the greatest error, which makes it difficult to assess by direct comparison. They use the same function for all the particles regardless of the difference in the intensity distribution of different particles, which may be the reason why they report such a large error.

Recently, Jaqaman *et al.* [13] and Serge *et al.* [14] independently developed 2D methods that are similar to each other for particle localization in cells. In cells, the size of a particle

is usually below or near the diffraction limit, and appears as a bright spot of a point-spread function (PSF) in an image so that its intensity distribution can be well-approximated by a 2D Gaussian. Thus, they fit the local intensity distribution around each local intensity maximum to a 2D Gaussian to obtain the feature location at subpixel resolution. As particles move close together, or one particle moves below another, the PSF's overlap and individual particles cannot be detected by fitting to a single Gaussian. To identify particles whose PSF's overlap, Jaqaman *et al.* fit the intensity distribution with multiple Gaussians and compare the residue. Serge *et al.* fit the primary peak with a Gaussian and subtract the fitted Gaussian from the image to find a second primary peak and repeat the process to identify signals that are above the noise floor in the image. This method is termed "deflation" in [14]. The idea is fundamentally the same as the tracking method we developed and present here of applying multiple passes of the Crocker and Grier tracking algorithm to an entire time series: that certain tracks (or particles) once successfully located and tracked, are removed from the raw data, then another pass is performed. Hence our multi-pass tracking is formally equivalent to the "deflation" process of Serge *et al.*, but is applied to non-overlapping colloids to unambiguously identify motions in time.

Jaqaman *et al.* also add some additional sophistication to their tracking process. One enhancement is the ability to handle particles appearing or disappearing, a challenge frequently met with in the cell environment and more acute in their 2D conditions due to particles moving in/out of the focus; and merging and splitting of particle trajectories are frequent [13]. Different costs associated with these events are assigned. These authors also provide models for the motion of individual particle based on their motional history and intensity, and optimize for an efficient algorithm by finding the most probable combination that gives them the highest fidelity of their tracks. Our algorithm is an adaptation of the Crocker and Grier particle tracking algorithm with a multi-pass of it. The Crocker/Grier algorithm deals with conditions of a particle appearing or disappearing by assigning a penalty that corresponds to the largest cutoff distance allowed for searching for a particle to belong to the same track [5], while Jaqaman *et al.* take as the penalty  $1.05 \times$  the maximum cost of all previous links. There is no fundamental difference here. In the Crocker and Grier algorithm, one sets the maximum number of consecutive steps a particle can miss by empirical experience. In 3D particle tracking in colloids, we do not permit disappearance of particles at all; or allow them to miss only one consecutive step before reappearing, in order to retain more tracks near the edge of the field of view. If particles are missing in more steps, there is no basis to believe they belong to the same track since we have no information for nearby particles that are outside of the field of view. We interpolate tracks linearly whenever there is a missing step. Statistically, this tends to bias towards more small steps, thus we would prefer a long track be broken into short tracks, than connect them with some uncertainty and bias the results to small steps. Hence, both our and the Jaqaman *et al.* tracking codes have sophisticated ways of handling particles being created and destroyed. Our multi-pass tracking code has the intelligence of the deflation in the Serge *et al.* code, as discussed above. Therefore, at present these methods do not provide any gain to us.

We have carried out direct comparison of the performance of all these localization methods for colloids that achieve subpixel 3D resolution. The test was created as follows: A simulation was run of dense particles, in 3D. This yields a set of the  $x, y, z$  positions. From this information, a stack of 2D images, simulating the appearance of real confocal microscope data, was created. This involved projecting the particle centers at the proper  $z$ -height, multiplying by a reasonable optical transfer function, and adding some noise. This generated a simulated raw data set that can be used by all the algorithms.

The results of this direct performance test show that our algorithm with multiple position refinement iterations achieves 11 nm resolution in 3D, for the simulated  $1 \mu\text{m}$  diameter parti-

Table 1. Performance data for accuracy in particle localization for each set of centroid-based 3D feature finding code. The results obtained show that our algorithm is at least half an order of magnitude better for accuracy in 3D.

Gao and Kilfoil		
root mean square error	in $\mu\text{m}$	in pixels
$r$	0.011	
$x,y$	0.0035	0.088
$z$	0.0097	0.049

Crocker and Grier		
root mean square error	$\mu\text{m}$	pixels
$r$	0.064	
$x,y$	0.028	0.70
$z$	0.050	0.25

Lu <i>et al.</i>		
root mean square error	$\mu\text{m}$	pixels
$r$	0.053	
$x,y$	0.025	0.625
$z$	0.040	0.20

Table 2. Performance data for time required to run each set of code on the image stack. The Lu *et al.* code is compiled in C++, the Crocker and Grier code is run in IDL, and the Gao and Kilfoil code is run in Matlab. For this test, the latter two were run on the same PC (2 GB RAM), and the first on a PC of equivalent capabilities.

time (s)	bpass	no bpass
Gao and Kilfoil	66	32
Crocker and Grier	36	2.4
Lu <i>et al.</i>	1	

cles; or, equivalently, 1/11 pixel in the  $x$  and  $y$  directions and 1/20 pixel in  $z$ . This is consistent with our earlier estimate of 10 nm accuracy obtained for experimentally determined particle positions [22] for 1.33  $\mu\text{m}$  diameter colloidal particles under experimental conditions similar to those modeled here. The results for the other algorithms were 53 nm resolution in 3D for the Lu *et al.* algorithm, consistent with an initial conjecture of 50 nm by that author, and 64 nm for the Crocker/Grier code.

To facilitate comparison across different optical systems, magnifications, and particle sizes, we also quote the error results in pixels. Roughly speaking, the Lu *et al.* algorithm is accurate to about a fifth of a pixel in  $z$  and the Crocker and Grier algorithm is accurate to about a quarter of a pixel in  $z$ , compared to roughly a twentieth of a pixel in  $z$  for our algorithm. Hence, all of three of these algorithms provide “subpixel resolution in  $z$ ”; however, the particle positions output by the Crocker/Grier and Lu *et al.* algorithms will suffer from pixel biasing in  $z$ , as we have demonstrated for the Crocker and Grier code. In particular that algorithm will continue to be unsuitable for extracting 3D quantities from the particle positions. With our code, this is no longer an issue.

We also have compared  $g(r)$  obtained from different feature finding algorithms to the actual



data (not shown). Our result is almost identical to the actual data while the other algorithms shows a radial distribution function with a lower first nearest neighbor peak. It is reasonable to say that the results for  $g(r)$  are essentially identical with the exception of the height of the first peak, which our code allows quantitatively to be measured, while other codes do not. Therefore plotting the first neighbor peak height of  $g(r)$  from experimentally-determined particle positions is not legitimate unless quantitative accuracy for particle positions can be demonstrated.

The runtime performance data for each set of code is presented in Table 2. Of the three algorithms, the Lu *et al.* code is easily more than an order of magnitude faster than the Crocker and Grier algorithm. In the test, the Lu *et al.* algorithm requires 1 second to complete the particle localization on the image stack. The Crocker and Grier feature finding algorithm is composed of two parts: an image filtering process and a particle localization process. The filtering process required 24 seconds and the particle localization process 2.4 seconds. Our algorithm runs the same filtering process but takes longer, 32 seconds, for the localization process due to the multiple (here, 20) iterations it performs for position refinement. In total, the Gao and Kilfoil algorithm with subpixel resolution in  $z$  following 20 position refinement iterations requires approximately twice the time the Crocker and Grier algorithm does for the entire feature finding process. The Lu *et al.* algorithm includes the steps that are separately listed as part of “bpass” in Crocker and Grier, so the time is reported under the “bpass” column. In [15], the positions of 100 million particles were located by Lu *et al.*. With our algorithm, we locate anywhere from 200 to 20000 particles depending on the experiment in each stack of a time series of typically 50 to 150 image stacks. The upper end of this, for 50 time stacks, requires running overnight on a PC and yields 1 million particles. Our code thus can locate  $10^8$  particles in three months on a single PC using uncompiled code, a reasonable amount of time for a publication, but atypical for our experiments where we focus on dynamics of several hundred to a thousand particles. The algorithms of Jaqaman *et al.* [13] and Serge *et al.* [14] developed for biological cells appear to be designed to extract on the order of 10 features in a cell. They are unlikely to be able to locate the positions of 100 million particles in a reasonable time, however direct performance tests are not possible. We might expect that with their sophisticated methods for modeling the particle motion to compensate for overlapping particles while all the 3D codes developed for colloids do not have overlapping particles, their algorithm would be more demanding than ours; however, direct comparisons are not meaningful between 2D and 3D methods.

In summary, all the algorithms presented have their relative strengths and weaknesses. For colloid physics, both the Lu *et al.* algorithm and our algorithm represent major advances in performance for 3D particle location over the Crocker and Grier code. The Lu *et al.* algorithm is optimized for speed, but is significantly less accurate than our code. By simply weighting the integrated intensity of a 3D distribution to a 2D spot, this code slightly outperforms the Crocker and Grier code for accuracy, which is all the more impressive given the Lu code’s speed and the expected trade-off in accuracy. Our algorithm is optimized for accuracy and is superior in accuracy by half a decade over all other algorithms in 3D; however, is 70 times slower than Lu’s if the filtering step and 20 iterations of position refinement are carried out. This is more food for thought than a particular suggestion that I feel strongly about. The multi-step tracking method we present here is optimized for dense systems with heterogeneous dynamics, and is intended for 3D data which means there is intrinsically less need to handle particles being created or destroyed. Using as it does the Crocker and Grier tracking code as a base, it is designed to handle these birth and death events; however, generous allowance for these events in dense systems will compromise the faithfulness of tracks. For the biological cell context, the algorithms of Jaqaman *et al.* and Serge *et al.* are in 2D and are optimized for small particles in the diffraction limit whose PSF’s can overlap and whose trajectories can cross. The Jaqaman *et al.* code handles appearance or disappearance of particles, and optimizes for an efficiency.

## 5. Experimental methods

The data for all the analysis of experimental data used to demonstrate the methods presented here was obtained on one dense colloidal sample prepared with depletion attraction, with the exception of data from an additional dense colloidal sample added in Fig. 6 for comparison. The colloids we use are fluorescently-labeled polymethylmethacrylate (PMMA) spheres. We add a linear polystyrene to induce a depletion attraction. The colloids sit in a mixed solvent of decahydronaphthalene (decalin), tetrahydronaphthalene (tetralin), and cyclohexyl bromide (CXB) that allows for an independent adjustment of the refractive index and relative buoyancy of the particles. We keep the refractive index matched in all samples. We use a VisiTech VT-Eye confocal fluorescence laser scanning microscope to acquire stacks of images (VisiTech International). For Fig. 2, 3(a) and 7, the full scanning volume is  $(45.3 \times 45.3 \times 73) \mu\text{m}^3$  at  $0.2 \mu\text{m}$  spacing in  $z$  and volume fraction  $\phi = 0.416$ . For Fig. 3(b) and 9, the field of view is  $(22.6 \times 22.6 \times 10) \mu\text{m}^3$  at  $0.2 \mu\text{m}$  spacing in  $z$  and  $\phi = 0.429$ . For Fig. 5 the field of view is  $(22.6 \times 22.6 \times 10) \mu\text{m}^3$  and  $\phi = 0.370$ . For the data represented by brown circles in Fig. 6, the field of view is  $(22.6 \times 22.6 \times 10) \mu\text{m}^3$  and  $\phi = 0.440$ . In the sample used for all of the above experiments the colloid diameter  $\sigma$  is  $1.33 \mu\text{m}$ , the polymer radius of gyration  $R_g$  is  $91.7 \text{ nm}$ , the depletion potential at contact  $U_{\text{dep}}(r = \sigma)$  is  $-2.86 k_B T$ , and the range of potential (set by  $\xi = 2R_g/\sigma$ ) is  $0.14 \sigma$ . For the data added for comparison in Fig. 6 represented by blue triangles and red squares, the colloid diameter  $\sigma$  is  $1.24 \mu\text{m}$ , the polymer radius of gyration is  $36.1 \text{ nm}$ , the depletion potential at contact is  $-3.02 k_B T$ , and the range of potential is  $0.058 \sigma$ . The data is acquired in imaging volume  $(22.6 \times 22.6 \times 10) \mu\text{m}^3$  and  $\phi = 0.492$ .

To create the simulated data for demonstrating the increased fidelity with the multipass tracking, configurations of equal size hard spheres at a dense volume fraction are generated based on the Clarke and Wiley algorithm [29]. This algorithm starts with a randomly distributed and overlapping configuration with a slightly larger volume fraction than the value initially set. Particles are then moved according to the vector sum of overlap to eliminate any overlaps until no more movement can be made. The particle radii are then decreased, particles are given a small random walk, the particle radii are increased, and the movement procedure is begun again. The entire process is repeated until overlap is within an acceptable range and the configuration reaches the desired volume fraction. The system is then equilibrated by standard molecular dynamics.

To create the simulated confocal microscopy data set for direct comparisons of the 3D particle location accuracy, we begin with one of the hard sphere simulation configurations. The box size for the simulation is 1 in the simulation units and the particle diameter is 0.0951. We assume that each particle extends 25 pixels in the  $x$  and  $y$  direction and 5 pixels in the  $z$  direction. This corresponds to an experimental case in which the size of a particle is  $1 \mu\text{m}$  and the pixel size in  $x$  and  $y$  is  $0.04 \mu\text{m}$ , and  $0.2 \mu\text{m}$  in  $z$ . Then accordingly the 53 digitized images generated have size  $263 \times 263$  pixels in the plane. The intensity of a particle is distributed according to a 3D Gaussian centered at the exact location of a particle with the Gaussian widths set to be the same for  $x$  and  $y$  ( $0.24 \mu\text{m}$ ) and a slightly broader Gaussian for  $z$  ( $0.3 \mu\text{m}$ ). The Gaussian is truncated such that the distribution holds only within the size of the particle. To add the effect of optical sectioning in  $z$ , we consider the thickness to be  $0.38 \mu\text{m}$  (FWHM) for a  $25 \mu\text{m}$  slit using a lens of 100X magnification with a Numerical Aperture of 1.4 [30], which gives a Gaussian width of  $\sigma = \text{FWHM}/2.35 = 0.16 \mu\text{m}$ . The contribution from neighbouring layers is then set by this Gaussian. In the end, we add a mean background intensity and random background noise to the image such that the simulated images are at similar signal to noise ratio to our experimental ones. We define the resolution of the feature finding as the square root of the mean squared deviations upon comparison of the particle coordinates obtained from a feature finding algorithm to the known particle coordinates.

Matlab (The Mathworks, Natick, MA) is used for all algorithms and analyses.

## 6. Conclusion

For the determination of microscopic dynamics at the single particle level, it is impossible to overestimate the importance of high resolution locating and tracking of features. We have implemented full 3D subpixel resolution position finding for colloids using a fractional shift of the pixels in 3D in successive iterations of position refinement that represents a major advance in precision, is suitable for dense colloidal systems, and is completely robust against pixel biasing. To our knowledge, this is the most precise three dimensional subpixel localization code available anywhere for colloids, by at least half an order of magnitude. We generated a data set to make direct comparison between the algorithm presented here and the two other 3D localization algorithms that offer anything close in terms of precision, and have presented the results here. The accuracy of our algorithm as determined from the direct comparison test is at least 5X better than either of the other two. We make the code available as open source at [31]. We describe the details of the algorithm here, and have demonstrated how severely the accurate measurement of dynamical and static quantities from 3D data depend on precise position refinement. Much confocal microscopy of colloids is carried out at lower resolution in the axial direction and the pixel biasing is most severe in these cases. With our code, this will no longer be a limiting factor. We have demonstrated that even  $x$  and  $y$  coordinate data suffer from pixel biasing from the 3D feature finding algorithms without several iterations of position refinement, unlike the situation in 2D position location where the first iteration is typically sufficient. The in-plane MSD obtained from 3D data without 3D subpixel accuracy is shown to be unreliable at short lag times. Caution is advised when using results derived from positional data without proper refinement in all three dimensions.

Furthermore, we have presented a method of tracking that is superior to the standard approach used in colloid physics for dense particle scenarios, by virtue of handling heterogeneous dynamics with high fidelity. We have also shown that tracking that compromises the fidelity of the tracks misses the most mobile proportion of particles, particularly when heterogeneous dynamics are present. When making comparisons on the most mobile fraction in these experimental systems [4, 6], capturing the rare large excursions is essential. Because of the great deal of current interest in glasses and jammed systems, our method of multi-pass tracking should prove useful for application to dense colloidal systems, particularly if coupled with the full subpixel resolution of our 3D feature finding code.

We have carried out direct comparison of the accuracy of the Lu *et al.* and Crocker and Grier algorithms to ours and confirmed our assertions of accuracy in particle location of our code. On the same set of simulated confocal images, our algorithm provides a 3D resolution of approximately 10 nm, half an order of magnitude better than the other algorithms. In conducting this performance test we have created a standard simulated raw data set with known original simulated positions that can be used to assess performance of all the algorithms.

The accuracy we obtain together with the saturation in the degree of refinement with successive iterations of the position correction (Fig. 3) point to another general issue in just how accurately software can locate particles: is there a fundamental limit? The accuracy of roughly  $\lambda/20$  we obtain for the  $x$  and  $y$  dimensions is, interestingly, similar to the limit of resolution on sizing of objects from static light scattering, again  $\lambda/20$ , as determined, for example, in [32]. This suggests a brief discussion to highlight several issues.

The first is the fundamental limit of accuracy in locating particles at all, for instance due to physical factors. The resolution we achieve is well below the diffraction limit, and other limitations to the fundamental theoretical accuracy of locating particles become relevant. If particles are diffusing, ever so slightly, in their cages during the imaging time of a few tenths

of a second, the positional accuracy is limited. The typical distance  $L$  they diffuse during the imaging time  $t$  may be estimated from the diffusion coefficient  $D$  via  $L \sim \sqrt{Dt}$ , with  $D$  extracted as the long time diffusive limit slope of the mean squared displacement versus time.

A second issue is how closely software can achieve this limit. Precision may be limited by image noise, as there are statistical fluctuations in the number of photons detected by the camera even under ideal conditions. As discussed in [34], in practice, well-made cameras approach the physical limit of performance of approximately 5 nm in the image plane when the full dynamic range of the camera is utilized and the feature image is spread over a sufficient number of pixels to reasonably represent its brightness distribution.

In dense systems, particularly close to gel or glass transitions, the particles never diffuse their own radius over times of hundreds to thousands of seconds. From the 1D MSD plotted in Fig. 6, we extract  $2D = 10^{-5}a^2$  and  $D = 1.8 \times 10^{-18}m^2/s$ . A particle thus typically diffuses  $\sim 1$  nm during the few tenths of seconds imaging time, less than the positional error we report here. Thus, in the special case of a dense system close to the gelation or glass transition, the finite exposure time required for visualization does not limit the positional accuracy. We conclude that our software has reached the fundamental theoretical accuracy of locating particles by imaging and software. We do not expect room for future improvement in the codes, and expect that the current work is truly at the limit of what is possible with image processing.

Finally, we anticipate dovetailing of our fractional-shifting refinement and tracking methods with recent tracking methods developed for use in biology. The issues of temporary confinement from caging in the glass, with particles on occasion making larger excursions, is quite analogous to the free vs. confined motion shown in Fig. 5 of [14], and in Fig. 4 of [12]. Our methods, now applied to colloids, could improve the location and tracking of quantum dots in cells. A proposed method could look like the following: apply the multiple-Gaussian fit and mobility models of Jaqaman *et al.* to segregate the overlapping particle positions, apply our position refinement to the all features at all time steps where the dots are not overlapping, and re-run the tracking on the refined positions using the deflation method we outline here in order to resolve all the particle displacements. For work in living cells, we already perform fitting of the local intensity distribution of a diffraction limited spot in biological cells with a 3D Gaussian and add an additional iteration of the Gaussian fit that is a linear interpolation between neighboring pixels (similarly to what we do in the present paper), and have established that further positional accuracy is possible over fitting the local intensity distribution alone (unpublished data). The combined approach we propose could be carried out on confocal microscopy data, which, importantly, would extend the Jaqaman *et al.* methods [13] to 3D.

We further anticipate future performance tests directly comparing the methods developed for use in biology, using simulated confocal microscopy data typical of the cell environment.