Supplementary Information – “Light Microscopy at Maximal Precision”

Matthew Bierbaum,[*] Brian D. Leahy[*],[†] Alexander A. Alemi,[‡] Itai Cohen, and James P. Sethna
Department of Physics, Cornell University, Ithaca, NY 14853

(Dated: August 29, 2017)

[*] These two authors contributed equally to this work.
[†] Electronic address: `bdl48@cornell.edu`
[‡] Work done while at Cornell, currently affiliated with Google Inc.

Contents

I. OVERVIEW

In this supplemental material we describe the details of our method for extracting parameters from experimental confocal images at the highest resolution possible without modifying the microscope itself. To achieve maximal resolution, we build a generative model which aims to describe the value of every pixel in the experimental image. That is, we create simulated images by explicitly modeling every relevant aspect of image formation including particle positions and sizes, the location of dirt in the optics, amount of spherical aberration in the lens, and the functional form of the point spread function. We describe each of these model components in detail in Section III and how we decided on these particular components in Section IV. In order to fit this model to the experiment, we adjust all model parameters until the features present in the true experimental image are duplicated in the simulated one. We decide when the fit is complete and extract errors of the underlying parameters by using a traditional Bayesian framework which is described in general terms in Section II. This high dimensional optimization is in general very difficult and so we describe our algorithmic improvements and particular techniques in Section V. Finally, we assess the accuracy of this method in extracting underlying parameters and compare its performance with traditional featuring methods in Section VI.

Overall, this document is meant to provide a roadmap for other researchers to follow when adapting this technique to other types of microscopy and other types of samples in order to extract the maximal amount of information from their experimental images.

II. BAYESIAN FRAMEWORK

When fitting a model to noisy data, it is useful to adopt a Bayesian framework in which we rigorously treat the noise as part of our model. In the case of our featuring method, we fit a model of each image pixel M_i to experimental data d_i, which can be described as a combination of signal and noise $d_i = S_i + \eta_i$. This noise is present due to the detection of a finite number of photons by the microscope sensor, noise in the electronics, etc. and can be well described for our system by uncorrelated $\langle \eta_i \eta_j \rangle = 2\sigma^2 \delta_{ij}$, Gaussian noise $\eta_i \sim \mathcal{N}(0, \sigma)$ (see Section III).

In a Bayesian framework, the likelihood that an individual pixel is correctly described by our model is given by the Gaussian likelihood,

$$\mathcal{L}(M_i \mid d_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-(M_i - d_i)/(2\sigma_i^2)} \tag{1}$$

For uncorrelated pixel noise, the entire likelihood of the model given the image is given by the product over all pixels, $\mathcal{L}(\mathbf{M} \mid \mathbf{d}) = \prod_i \mathcal{L}(M_i \mid d_i)$. We are ultimately interested in the probability of the underlying parameters given the image we record. According to Bayes' theorem, we can write this as

$$\begin{aligned} P(\boldsymbol{\theta} \mid \mathbf{d}) &\propto P(\mathbf{d} \mid \boldsymbol{\theta}) P(\boldsymbol{\theta}) \\ &\propto \mathcal{L}(\mathbf{M}(\boldsymbol{\theta}) \mid \mathbf{d}) P(\boldsymbol{\theta}) \end{aligned}$$

where $P(\boldsymbol{\theta})$ are priors that allow us to incorporate extra information about the parameters $\boldsymbol{\theta}$. These priors can be as simple as the fact that the particle radius is positive definite or that a group of images share similar PSFs. For example, an overlap prior $P_{\mathrm{overlap}}(\mathbf{x}_i, \mathbf{x}_j, a_i, a_j) = \mathrm{H}(a_i + a_j - |\mathbf{x}_i - \mathbf{x}_j|)$, where H is the Heaviside step function, can be used to impose the physical constraint that particles cannot overlap. However, we found that the overlap prior only becomes relevant when the free volume of a particle is small compared to the average sampling error volume (when a particle is caged by ~ 1 nm on all sides) and so we ignore it most of the time.

We primarily work with the log-likelihood function $\log \mathcal{L}$ because the number of pixels in the image can be very large, on the order 10^7. For Gaussian noise, the log-likelihood is precisely the square of the L_2 norm between the model and the data. Therefore, we are able to maximize this log-likelihood using a variety of standard routines including linear least squares and a variety of Monte-Carlo sampling techniques. After optimizing, we can determine errors with one of two methods. First, we can use estimate the covariance matrix of the errors by from $J^T J$, where $J_{i\mu} = \partial_\mu r_i$ is the derivative of the residuals with respect to the fit parameters. (*i.e.* the Jacobian of the transformation from parameter space to data space). Second, we can use standard Monte-Carlo algorithms to sample from the posterior probability distribution to extract full distributions of the model parameters.

In this way, any quantity of interest that is a function of particle distribution can be calculated using Monte-Carlo integration by

$$\begin{aligned}\langle \mathcal{O}(\boldsymbol{\theta}) \rangle &= \int \mathcal{O}(\boldsymbol{\theta}) P(\boldsymbol{\theta} \mid \mathbf{d}) \, \mathrm{d}\boldsymbol{\theta} \\ &= \frac{1}{N} \sum_{i}^{N} \mathcal{O}(\boldsymbol{\theta}^{i})\end{aligned}$$

Here, $\boldsymbol{\theta}^i$ is a parameter vector sampled fairly from the posterior probability distribution and $\mathcal{O}(\boldsymbol{\theta}^i)$ is an observable such as the pair correlation function, packing fraction, or mean squared displacement. Calculating higher-order moments provides estimated errors and error correlations on these observables. This is one of the more powerful aspects of this method – one can generate a probability distribution for each parameter and directly apply these distributions to any observable that can be inferred from the parameters.

Given this Bayesian framework, the main idea of this work is to create a full generative model for confocal images of spherical particles and provide algorithmic insights in order to implement the model on commodity computer hardware.

III. GENERATIVE MODEL

Most of the difficulty in our method lies in creating a generative model that accurately reproduces each pixel in an experimental image using the fewest number of parameters possible. Our model is a physical description of how light interacts with both the sample and the microscope optics to create the distribution of light intensity that is measured by the microscope sensor and rendered as an image on the computer. In this section, we describe the model which we use to generate images similar to those acquired by line-scanning confocal microscopy of spherical particles suspended in a fluorescent fluid.

Our generative model aims to be an accurate physical description of the microscope imaging; it is not a heuristic. Creating this model requires a detailed understanding of image formation of colloidal spheres in a confocal microscope. In the simplest view, our samples consist of a continuous distribution of dye distributed throughout the image. If the fluid is dyed (as for the images in this work), due to diffusion the dye is uniformly distributed through the fluid. The fluid-free regions, such as those occupied by the particles, are perfectly dye-free. The sample is illuminated with a laser focused through an objective lens. This focused laser excites the fluorescent dye only in the immediate vicinity of the lens's focus. An objective lens captures the dye's emitted light, focusing it through a pinhole to further reject out-of-focus light. The collected light passes through a long-pass or band-pass filter, which eliminates spurious reflected laser light before collection by a detector. This process produces an image of the sample at the focal point of the lens. Finally, rastering this focal region over the sample produces a three-dimensional image of the sample.

However, the actual image formation is more complex than the simple view outlined above. Excessive laser illumination can cause the dye to photobleach. Due to dirt and disorder in the optical train, the sample is not illuminated uniformly. Diffraction prevents the laser light from being focused to a perfect point and prevents the objective lens and pinhole from collecting light from a single point in the sample. Aberrations are present if the sample's refractive index is not matched to the design of the objective lens, broadening the diffractive blur deeper into the sample. Both the illuminating and fluorescing light can scatter off refractive index heterogeneities in the sample due to the particles.

Some of these complications can be eliminated by careful sample preparation. In practice, we eliminate photobleaching by using an excessive amount of dye in our samples and illuminating with a weak laser light. We eliminate scattering by matching the refractive index of the particles to the suspending fluid – it is fairly easy to match the refractive indices to a few parts in 10^3. Since the scattering is quadratic in the index mismatch, the effect of turbidity due to multiple-scattering is very weak in our samples. However, the rest of these complications must be accurately described by the generative model.

Based on this physical setup, we can describe the confocal images through three main generative model components:

- *Platonic image* $\Pi(\mathbf{x})$ – the physical shape of the dye distribution in the sample (unmodified by perception of light).

- *Illumination field* $I(\mathbf{x})$ – the light intensity as a function of position, including both laser intensity variation from disorder in the optics and intensity attenuation into the sample.

- *Point spread function* $P(\mathbf{x}; \mathbf{x}')$ – the image of a point particle due to diffraction of light, including effects from index mismatch and finite pinhole diameter.

plus three minor additional fit model components:

- *Image Background* c, $B(\mathbf{x})$ – the overall exposure of the image c and the background values corresponding to a blank image without dye, B.
- *Rastering Step Size* z_{scale} – the displacement distance of the lens as it rasters along the optical axis.
- *Sensor noise* σ – the noise due to shot noise from finite light intensity reaching the sensor or electronic noise at the sensor.

These components are combined to form the image through convolution

$$\mathcal{M}(\mathbf{x}) = B(\mathbf{x}) + \int \mathrm{d}^3 x' \; [I(\mathbf{x'})(1 - \Pi(\mathbf{x'})) + c\Pi(\mathbf{x'})] P(\mathbf{x} - \mathbf{x}'; \mathbf{x}) \tag{2}$$

which is sampled at discrete pixel locations to give the final image $M_i = \mathcal{M}(\mathbf{x}_i)$.

Here, we describe each part of our model in detail along with our explanations and motivations behind any simplifications. In subsequent sections we will also discuss other aspects of image formation which may result in other model choices and why we omit them from the final form of the model.

A. Platonic image

The Platonic image must accurately represent the continuous distribution of fluorescent dye in the sample on the finite, pixelated image domain. The colloidal sample consists of a collection of spherical particles embedded in the solvent, with either only the particles or only the solvent dyed. Our Platonic image should then consist of the union of images of individual spherical particles, with their corresponding radii and positions. Thus, if we have a method to accurately represent one colloidal sphere, we can easily construct the Platonic image in our generative model.

A naïve way to generate the Platonic image of one sphere would be simply to sample the dye distributions at the different pixel locations, with each pixel being either 0 (if it is outside the sphere) or 1 (if it is inside the sphere) with no aliasing. This method will not work, since a pixel value in the Platonic image can only change when a sphere's position or radii has shifted by one pixel. This method of Platonic image formation would produce a generative model that does not adequately distinguish between particle locations separated by less than 1 pixel or 100 nm! Simply multiplying the resolution and corresponding coarse-graining of the boolean cut by a factor of N in each dimension increases the resolution of this method to $1/N$ pixels. However, calculating these high resolution platonic spheres is computationally expensive, requiring 10^9 operations to draw spheres capable of determining positions within 0.01 px.

To find the correct representation of a Platonic sphere, we examine the mechanism of image formation in Eq. 2. The final image results from a convolution of the Platonic image with the point-spread function $P(\mathbf{x} - \mathbf{x'}; \mathbf{x})$. Thus, we need a representation of a sphere that will produce the correct image after being convolved with the point-spread function. To do this, we recall that a convolution is a multiplication in Fourier space. However, creating the image of the sphere in Fourier space is problematic since there will be undesirable ringing in the Platonic image due to the truncation from the finite number of pixels (*i.e.* Gibbs phenomenon). Moreover, each update of one particle requires updating all the pixels in the image, which is exceedingly slow for large images.

Instead, we look for a functional form in real space that approximates the numerically-exact truncated Fourier series, where the truncation arises due to a finite number of pixels. For a sphere with radius a at position $\mathbf{p}$, this truncated Fourier series is given by $\tilde{\Pi}(\mathbf{q}; \mathbf{p}, a) = 4\pi a^3 (j_1(q)/q) e^{i\mathbf{q}\cdot\mathbf{p}}$, where $\mathbf{q}$ is sampled only at frequencies in the image. We can view the truncation operation as a multiplication in Fourier space by a boxcar $\mathrm{H}(1 - |q_x|)\mathrm{H}(1 - |q_y|)\mathrm{H}(1 - |q_z|)$, where $\mathbf{q}$ is the variable inverse to position, measured in px^{-1}. By the convolution theorem, this truncation corresponds to a convolution in real space with $\mathrm{sinc}(x)\,\mathrm{sinc}(y)\,\mathrm{sinc}(z)$, using the inverse Fourier transform of the boxcar as the sinc function. Thus, the numerically exact image of a sphere would be the analytical convolution of $\mathrm{sinc}(x)\,\mathrm{sinc}(y)\,\mathrm{sinc}(z)$ with a sphere of radius a at position p, represented on a discrete grid. However, the convolution with the sinc function is analytically intractable. To circumvent this, we approximate the sinc function by a Gaussian. This gives a representation of the correctly-aliased Platonic image $\Pi(\mathbf{x; a})$ of a sphere of radius a as

$$\Pi(\mathbf{x}) = \mathcal{S}(\mathbf{x}) * \left[\left(2\pi\sigma_x^2\sigma_y^2\sigma_z^2\right)^{-1/2} e^{-x^2/2\sigma_x^2} e^{-y^2/2\sigma_y^2} e^{-z^2/2\sigma_z^2} \right] \tag{3}$$

where $\mathcal{S}(\mathbf{x}; \mathbf{p}, a) = H(|\mathbf{x} - \mathbf{p}| - a)$ where $H(x)$ is the Heaviside step function, which is either 0 or 1 depending on whether $|\mathbf{x} - \mathbf{p}| > a$ or $< a$, and $*$ denotes convolution. The Gaussian widths σ should be approximately 1 px; however, if the ratio of the z pixel size to the xy pixel size $z_{\text{scale}} \neq 1$, then σ_z will not be the same as σ_x and σ_y.

While Eq. 3 does not generally admit a simple solution, there is a closed-form functional form for the symmetric case $\sigma_x = \sigma_y = \sigma_z$. In the symmetric case ($z_{\text{scale}} = 1$) Eq. 3 takes the form

$$\Pi(\mathbf{x}) = \frac{1}{2}\left[\mathrm{erf}\left(\frac{a - r}{\sigma\sqrt{2}}\right) + \mathrm{erf}\left(\frac{a + r}{\sigma\sqrt{2}}\right) \right] - \frac{1}{\sqrt{2\pi}} \frac{\sigma}{r} \left[e^{-(r-a)^2/2\sigma^2} - e^{-(r+a)^2/2\sigma^2} \right] \tag{4}$$

where r is the distance from the particle's center. The first bracketed group of terms corresponds to treating the sphere as a flat surface, and the second bracketed group corresponds to the effects of the sphere's curvature on the integral. In each sub-grouping, the first term that depends on $r - a$ reflects the contribution due to the particle's nearer edge, and the second term that depends on $r + a$ reflects the contribution due to the particle's farther edge. We then fit σ in Eq. 4 to best match the exact Fourier space image of a sphere, giving a value $\sigma \approx 0.276$.

Although Eq. 3 does not admit a simple solution for $z_{\text{scale}} \neq 1$, we can use the exact form for $z_{\text{scale}} = 1$ to construct an approximate solution. Since both $\text{erf}(x)$ and e^{-x^2} approach their asymptotic values extremely rapidly, and since at the best fit $\sigma \approx 0.276$, the argument $(a + r)/\sigma \gg 1$ for even moderately small radii, we approximate the terms $\text{erf}((a + r)/\sigma\sqrt{2}) \approx 0.5$ and $\exp(-(r + a)^2/2\sigma^2) \approx 0$ to an excellent accuracy. We then write the position vector in terms of its direction from the sphere's center $\hat{\mathbf{x}}$ and a vector $\boldsymbol{\delta}\mathbf{x}$ as $\mathbf{x} \equiv a\hat{\mathbf{x}} + \boldsymbol{\delta}\mathbf{x}$; the vector $\boldsymbol{\delta}\mathbf{x}$ is the distance of the point from the sphere's surface, and the vector $a\hat{\mathbf{x}}$ is the location on the sphere's surface closest to the point $\mathbf{x}$. Finally, we replace $(a - r)/\sigma$ in Equation (4) by $\delta = \sqrt{(\delta x/\sigma_x)^2 + (\delta y/\sigma_y)^2 + (\delta z/\sigma_z)^2}$, where δx is the projection of $\boldsymbol{\delta}\mathbf{x}$ on the i^{th} coordinate axis and σ_i is the pixel blurring along each direction; $\sigma_z = z_{\text{scale}}\sigma_x$ for our geometry with identical x and y pixel sizes and a varying z pixel size. Combining all these produces the following equation for a Platonic sphere:

$$\Pi(\mathbf{x}) = \frac{1}{2}\left[1 + \text{erf}\,(\delta)\right] - \frac{1}{\sqrt{2\pi}}\frac{\sigma}{r}e^{-\delta^2/2} \tag{5}$$

Note that this approximation is exact in the limit of infinite sphere radii. Empirically, we find that this approximation works quite well, giving differences in the Platonic image of a few percent from a numerical solution to Eq. 3 as well as high resolution boolean cut real-space spheres (see Fig. 1).

While this implementation of the Platonic image correctly captures most of the effects of finite-pixel size, there are still some minor details that need to be fixed to give unbiased images. By construction, Eq. (4) conserves volume – its integral over all space is $4/3\pi a^3$ since the Gaussian kernel is normalized. However, when $\Pi(\mathbf{x})$ is sampled on a pixelated grid, its sum is not exactly $4/3\pi a^3$ but is slightly different, depending on the position of the particle's center relative to a voxel's center. The slight change in volume is important for two reasons. First, the convolution with the PSF in our image generation (see next subsection) suppresses high-frequency portions of the image, but it does not affect the $\mathbf{q} = \mathbf{0}$ component, *i.e.* the image sum or the particle volume. Since we aim to create a Platonic image that accurately represents the final image, we need the $\mathbf{q} = \mathbf{0}$ component of the Platonic image to be correct. Secondly, as discussed in section IV the real microscope image is actually an integral over a finite pixel area. As such, the image recorded on the detector preserves the particle's volume or the $\mathbf{q} = \mathbf{0}$ component of the image. To circumvent this issue of incorrect particle volume, instead of drawing the particle at its actual radius we draw it with a slightly different radius that preserves the particle's volume, which we accomplish with an iterative scheme. The results of this iterative scheme are shown in Fig. 1 along with the errors it introduces. Incidentally, the effects of image pixelation on image moments higher than $\langle 1 \rangle$, *e.g.* $\langle \mathbf{x} \rangle$ and its effects on the particle positions, are much smaller than the noise floor in our data at a moderate SNR (see section IV).

The representation in equation 4 is the best method for forming Platonic spheres on a pixelated grid that we have found. However, there are other, simpler methods which work almost as well as the Platonic sphere. Aside from the important curvature term, equation 4 is basically an erf() interpolation between particle and void at the particle's edge. Other interpolation schemes can provide similar results. For instance, the spheres could be constructed by ignoring the curvature term and replacing the erf with a logistic $1/(1+\exp((r-a)/\alpha))$, a linear interpolation between particle and void at the pixel edge, or a cubic interpolation at the pixel edge. We have also implemented these methods for generating Platonic images of spheres, fitting the parameters to match the exact Fourier representation. For the logistic we fit α, for the linear interpolation we fit the slope, and for the cubic we fit one parameter and constrain the other two such that the Platonic image and its derivative are continuous. While all of these methods are functional, they are not significantly faster than the exact Gaussian approximation in equation 4 and result in slightly worse featuring errors (see table I). As a result, we use the exact Gaussian approximation, but include these other options in our package for ease of use with more complicated shapes where the integral in equation 3 might not be analytically tractable.

The Platonic image needs to represent accurately all objects in the image, not just the spheres. In particular, when the solvent is dyed, the image usually contains a dark coverslip or its shadow from the point-spread function. We model this dark coverslip as a slab occupying a half-space. The slab is characterized by a z-position and by a unit normal $\hat{\mathbf{n}}$ denoting the perpendicular to the plane. To capture accurately sub-pixel displacements of the slab, we use the image of a slab convolved with a Gaussian as above for a sphere; for the slab this gives a simple error (erf) function.

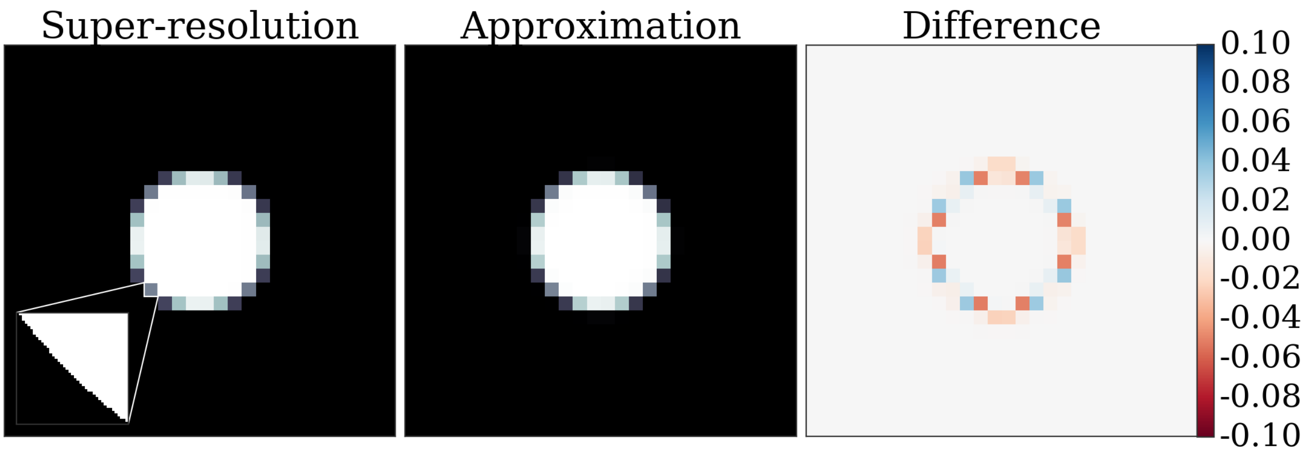

FIG. 1: **Platonic sphere generation**. A comparison of our approximate platonic sphere generation method to a sphere created by performing a boolean cut $\Pi(\mathbf{x}) = \int_{\text{pixel}} \mathrm{d}\mathbf{x}' H(|\mathbf{x}-\mathbf{x}'-\mathbf{p}|-a)$ on a lattice 100× higher in resolution in each dimension compared to the final image. On the left we show the super resolution sphere with fractional volume error $\delta V/V = 10^{-6}$ and an inset displaying the jagged edges caused by discrete jumps in distance. This is in contrast to the iterative approximate platonic sphere with volume error $\delta V/V = 10^{-16}$ drawn at an effective radius with change $\delta a/a = 2\times 10^{-4}$. The differences between individual pixels along the center of the sphere (right panel) show a high frequency structure with a maximal relative value 0.08. These high frequency features are dramatically reduced later in the image formation process through the convolution with the point spread function.

B. Illumination field

In order to illuminate the sample, confocal microscopes scan a laser over the field of view using several distinct patterns including point, line, and disc scanning. This illumination laser travels through the optics train and interacts with fluorescent dye in the suspension causing it to emit light in a second wavelength which is then detected. The intensity of this illumination pattern depends on the aberrations in the optics, variation of intensity due to photo-bleaching, as well as dirt in the optical train which creates systematic fluctuations in illumination across the field of view. Accounting for these variations is important as they can account for most of the intensity variation in an image. In the case of our line scanning confocal microscope, these patterns manifest themselves as stripe patterns perpendicular to the scan direction, as the line-scan drags dirt across the field of view, overlaid on aberrations and optical misalignments which cause the corners of the image to dim.

Confocal microscopes image by rastering in z, illuminating each xy plane separately. Ideally, the microscope illuminates each plane identically. In practice, aberrations due to refractive index mismatches cause a dimming of the illumination with depth into the sample [1]. Since this overall dimming only depends on the depth z from the interface and not on the xy position in the sample, it is natural to describe the illumination field as a product of an xy illumination and a z modulation:

$$I(\mathbf{x}) = I_{xy}(x,y) \times I_z(z) \quad . \tag{6}$$

Empirically we find that illumination fields of this form can accurately describe our real confocal images, without incorporating any coupling between xy and z.

We describe each of the separate functions I_{xy} and I_z by a series of basis functions. Since the modulation in z is fairly smooth [1], we describe $I_z(z)$ by a polynomial $P_z(z)$ of moderate order $\approx$ 7-11 for 50-70 z-slices; typically we use a Legendre polynomial as the orthogonality accelerates the fitting process. The in-plane illumination of a confocal is determined by its method of creating images. Our confocal is a line-scanning confocal microscope, which operates by imaging a line illumination parallel to the x axis and simultaneously collecting the line's fluorescent image. This line is then scanned across the image in y. As a result of this scanning, any dirt in the optics is dragged across the field of view, creating the illumination with stripes along the x-direction visible in Fig. 2. To model these stripes, we treat the variation along x and y differently. We write the xy illumination field as

$$I_{xy}(x,y) = \sum_k B(x;\mathbf{c}_k) \times P_k(y) \quad , \tag{7}$$

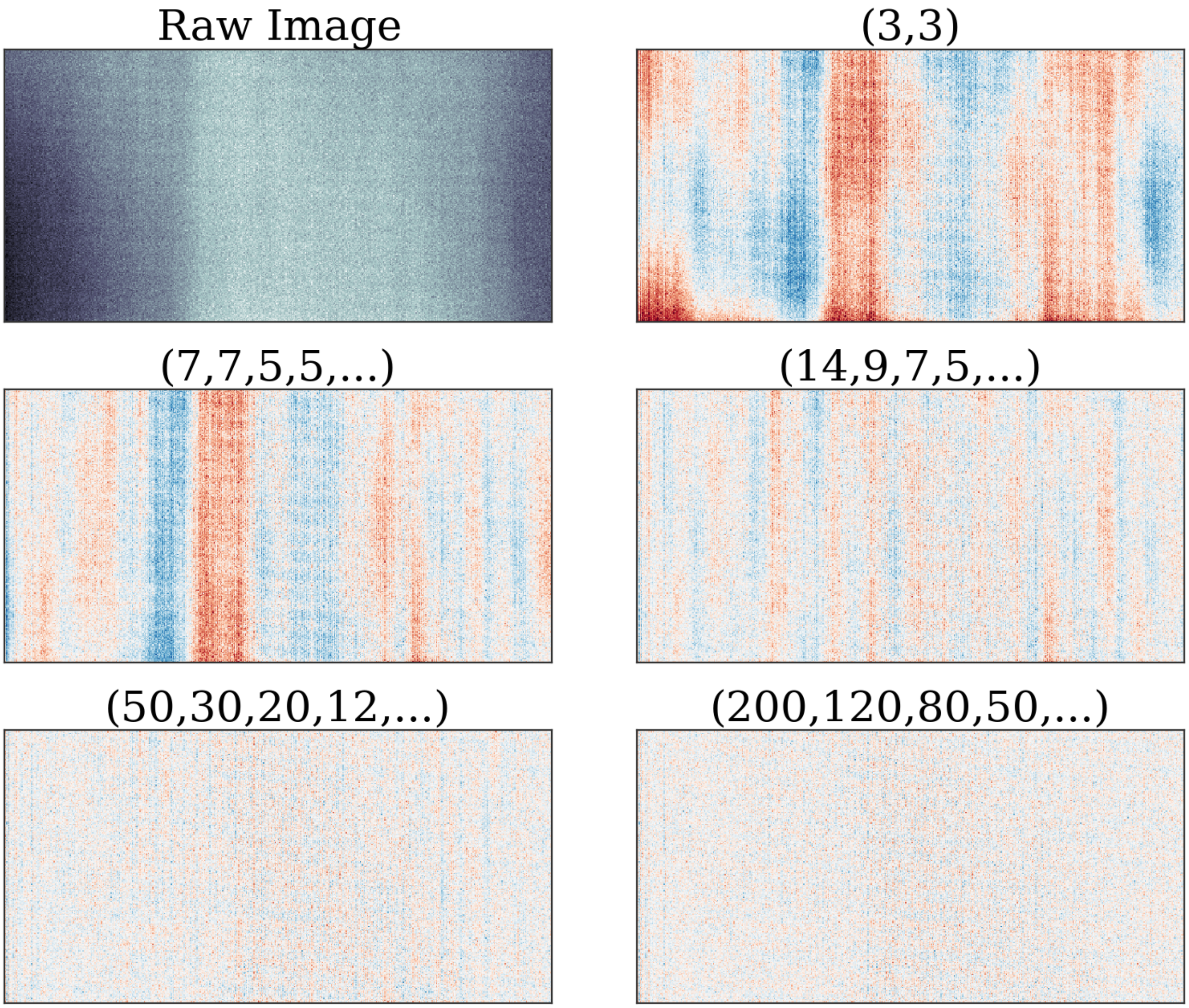

FIG. 2: **Illumination field residuals**. A blank confocal image and its fit to the Barnes ILM in equation 8 over varying number of coefficients. Fitting the illumination with a low-order ILM of $(3, 3)$ Barnes points removes the large fluctuations over the image but clearly shows stripes in the image. The notation $(n_0, n_1, n_2, ...)$ corresponds to a Barnes ILM with n_0 coefficients in the expansion for $P_0(y)$, n_1 coefficients for $P_1(y)$, etc. Increasing the number of points to $(7, 7, 5, 5, 5)$ or $(14, 9, 7, 5, 5, 5)$ removes the overall modulation in y but leaves clear stripes in the image. Only at high orders of $(50, 30, 20, 12, 12, 12, 12)$ or $(200, 120, 80, 50, 30, 30, 30, 30, 30, 30, 30)$ do these stripes disappear. The residuals shown in the figure are all at the same scale and are averaged over the image z for clarity.

where $B_k(x; \mathbf{c}_k)$ is a Barnes interpolant in x and $P_k(y)$ a Legendre polynomial in y. Barnes interpolation is a method of interpolating between unstructured data using a given weight kernel [2], similar to inverse distance weighting, using a truncated Gaussian kernel to allow for strictly local updates to the high frequency illumination structure. We use an interpolant with equally spaced anchor points in x throughout the (padded, see section III C) image. The k^{th} Barnes interpolant has a large number of free parameters, described by the vector $\mathbf{c}_k$; the size of $\mathbf{c}_k$ is equal to the number of anchoring points in the Barnes. To account for the fine stripes in the image, we use a large number of points for the Barnes associated with low-order polynomials, and decrease the number of points for higher-order polynomials. For a typical image of size $(z, y, x) = (50, 256, 512)$ pixels, we use coefficient vectors of length $(\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4, \mathbf{c}_5, \mathbf{c}_6, \mathbf{c}_7, \mathbf{c}_8, \mathbf{c}_9, \mathbf{c}_{10}) \approx (200, 120, 80, 50, 30, 30, 30, 30, 30, 30, 30)$. While this is a large number of coefficients, there are orders of magnitude fewer coefficients than pixels in the image. As a result, all of the ILM parameters are highly constrained (on the order of a few parts in 10^5, varying wildly with the parameter), and we do not overfit the image.

Putting this all together, we use an ILM given by:

$$\left[\sum_k B_k(x;\mathbf{c}_k)P_k(y)\right]\left[\sum_j d_j P_j(z)\right] \quad . \tag{8}$$

This ILM accurately describes measured confocal illuminations, as determined both from blank images and from images with colloidal particles in them. While the Barnes structure of this ILM is optimized for line-scanning microscopes, it can easily be changed. For ease of use for different microscopes or imaging modalities we have implemented various ILMs consisting of simple Legendre polynomial series, as functions $P_{xy}(x,y) \times P_z(z)$, $P_{xy}(x,y) + P_z(z)$, and as $P_{xyz}(x,y,z)$. Other illumination structures – such as a radially or azimuthally striped ILM for spinning-disk confocals – could also easily be incorporated into PERI's framework.

How well do these functional forms fit to experimental data for a line-scanning confocal microscope? We acquire blank images of a water-glycerol mixture as a function of depth and fit these data with Barnes illuminations of the form Eq. 8. As a function of the number of Barnes points in x and the polynomial degree in y, we look at the magnitude and patterns of the residuals. In Fig. 2, we see large scale structure in the ILM residuals, suggesting that high-order polynomials and Barnes interpolants with a large number of points are necessary. Fitting out the low-order background reveals the find stripes in x emerge due to the line-scan nature of our machine. Finally, at higher orders of interpolants and polynomials we are able to adequately capture all illumination variation independent of depth into the sample.

Fitting the ILM correctly is essential for finding the correct particle positions and radii. Fig. 3 demonstrates the effect of featuring a real confocal image with an illumination field of insufficient order. In the left panel is an image featured with a high-degree polynomial illumination of 9th order in the x-direction and of 5th order in the y- and z-directions. While these polynomials are high-order, they are not high enough to capture all of the structure in the light illumination. There is a clear bias in the featured radii, with particle radii being systematically larger on the edge of the image and smaller in the middle. These biases arise from large stripes in the confocal illumination due to the line-scanning nature of our confocal. Using a higher-order 25th degree polynomial in the x-direction (upper right panel) eliminates the effect of these stripes, as visible in the featured particle radii plotted as a function of x in the bottom panel. Note that the particle radii may be biased by as much as 1 px or 100 nm due to effects of the spatially varying illumination field.

C. Point spread function

Due to diffraction, the illuminating laser light focused from the microscope's lens and the detected fluorescent light collected from the sample are not focused to a single point. Instead, the light is focused to finite-sized diffraction-limited blur. To reconstruct an image correctly we need to account for the effects of diffraction in image formation.

A confocal microscope first illuminates the sample with light focused through the microscope lens. The lens then collects the light emitted from fluorophores distributed in the sample. As a result, the final image of a point source on the detector results from two separate terms: an illumination point-spread function P_{ilm} that describes the focusing of the incoming laser light, and a detection point spread function P_{det} that describes the focused fluorescent light collected from the emitted fluorophores. Since a fluorophore is only imaged if it is both excited by the laser illumination and detected by the camera, the resulting point-spread function for a confocal with an infinitesimal pinhole is the product of the illumination and detection point-spread functions: $P(\mathbf{x}) = P_{\mathrm{ilm}}(\mathbf{x})P_{\mathrm{det}}(\mathbf{x})$. For a confocal with a finite-sized pinhole, this product becomes a convolution over the pinhole area. The two separate point-spread functions (PSFs) P_{ilm} and P_{det} can be calculated from solutions to Maxwell's equations in the lens train [1, 3–5]. The PSFs can be written as integrals over wavefronts of the propagating light.

An additional complication arises from the presence of an optical interface. Most microscope lenses are essentially "perfect" lenses, creating a perfect focus in the geometric optics limit. However, refraction through the optical interface destroys this perfect focus and creates an image with spherical aberration. In addition, the refracted rays shift the point of least confusion of the lens from its original geometric focus. For a confocal geometry, this spherical aberration and focal shift depend on the distance of the nominal focal point from the optical interface z_{int}.

All of these effects have been calculated in detail by many previous researchers [1, 3–5]. The PSFs depend on several parameters: the wave vectors of the incoming and outgoing light k_{in} and k_{out}, the ratio of the indices of refraction $n_{\mathrm{sample}}/n_{\mathrm{lens}}$ of the sample and the optical train design, the numerical aperture of the lens or its acceptance angle α, and the distance focused into the sample z_{int}. For completeness, we repeat the key results here. In polar coordinates, the illumination PSF $P_{\mathrm{ilm}}(\rho,\phi,z)$ for illuminating light with wave vector k_{in} traveling through a lens focused to a

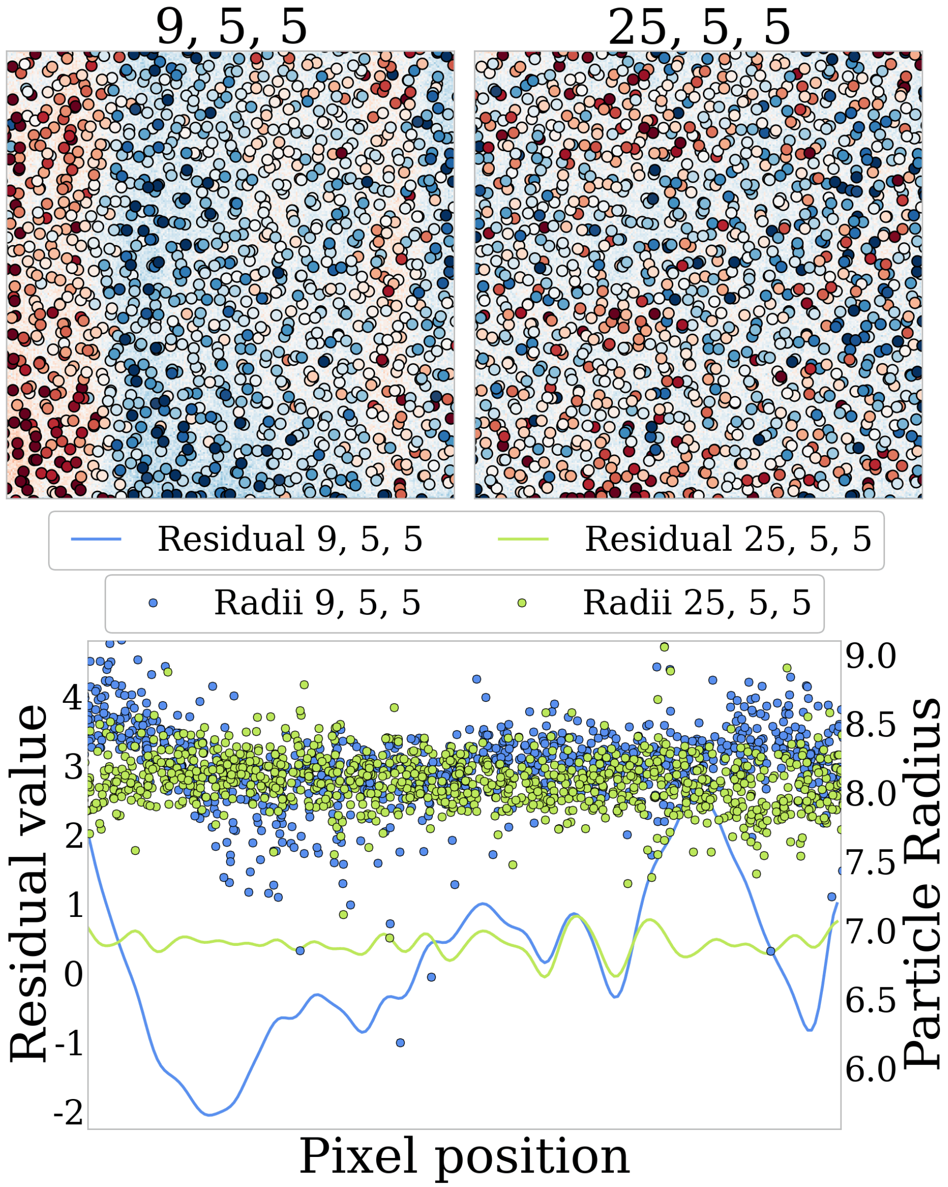

FIG. 3: **ILM generated biases**. Using an incorrect illumination field results in significant biases. The upper left panel shows an image featured with a $(9, 5, 5)$ order polynomial in (x, y, z). In the foreground are the featured particle radii, color-coded according to their difference from the mean. In the background is the residuals of the featured image. Clear stripes are visible in both the featured radii and the residuals. The particles are systematically much larger on the left side of the image, before decreasing in size in the middle and increasing again in a small stripe on the image's right side. In contrast, when the image is featured with a higher-order $(25, 5, 5)$ degree polynomial, shown in the upper right, these systematic residuals disappear. The bottom panel shows the particle radii and image residuals for the two illumination fields as a function of the image x direction.

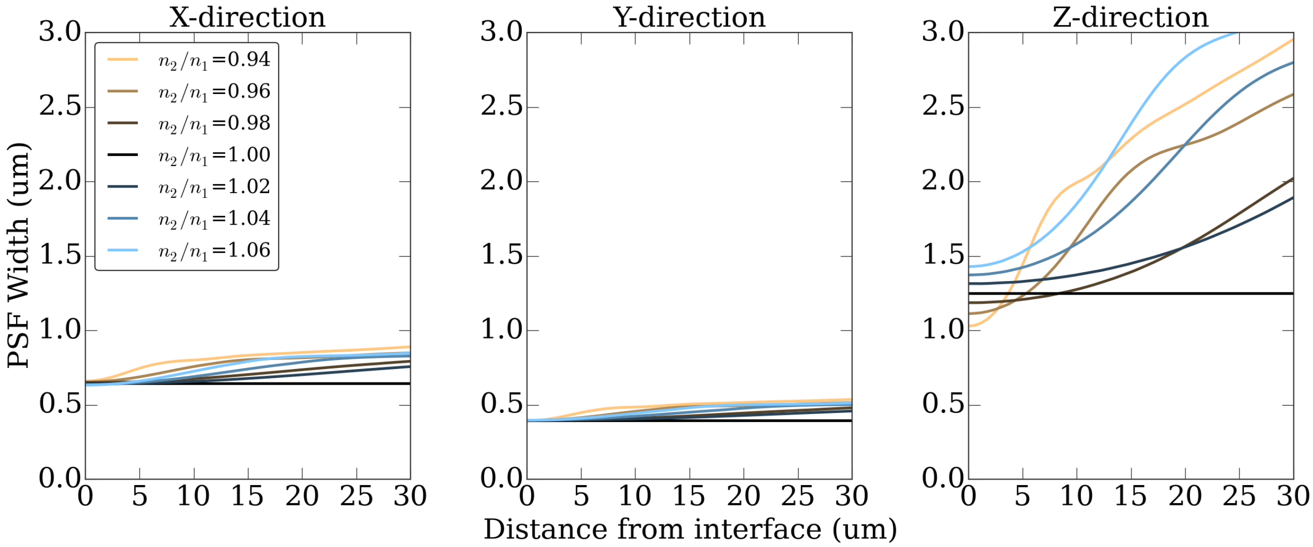

FIG. 4: **PSF widths vs depth**. The x (left panel), y (center panel), and z (right panel) widths of a line-scanning confocal's PSF with an NA=1.4 lens as a function of distance from the interface, for various refractive index mismatches. The width of the point-spread function generally increases with depth and with index mismatch due to increased spherical aberrations. The width is broadest in the z (axial) direction, and is narrower in the y direction than along the x direction of the line illumination. In each panel, the width plotted is the distance between the first and third quartiles of the PSF.

depth z_{int} from the interface is [1]

$$\begin{aligned}
P_{\text{ilm}}(\mathbf{x}) &= |K_1|^2 + |K_2|^2 + \frac{1}{2}|K_3|^2 + \cos 2\phi \left[K_1 K_2^* + K_2 K_1^* + \frac{1}{2}|K_3|^2 \right] \quad \text{, where} \\
\begin{pmatrix} K_1 \\ K_2 \\ K_3 \end{pmatrix} &= \int_0^\alpha \sqrt{\cos\theta'} \sin\theta' e^{-ik_{\text{in}} f(z,\theta')} \begin{pmatrix} \frac{1}{2}(\tau_s(\theta') + \tau_p(\theta')\cos\theta_2) J_0(k_{\text{in}}\rho\sin\theta') \\ \frac{1}{2}(\tau_s(\theta') - \tau_p(\theta')\cos\theta_2) J_2(k_{\text{in}}\rho\sin\theta') \\ J_1(k_{\text{in}}\rho\sin\theta')\tau_p(\theta')\frac{n_1}{n_2}\sin\theta' \end{pmatrix} d\theta' \\
f(\theta) &= z_{\text{int}}\cos\theta - \frac{n_2}{n_1}(z_{\text{int}} - z)\sqrt{1 - \left(\frac{n_1}{n_2}\right)^2 \sin^2\theta}
\end{aligned} \tag{9}$$

Here $\tau_s(\theta')$ and $\tau_p(\theta)$ are the Fresnel reflectivity coefficients for s and p polarized light, J_n is the Bessel function of order n, and θ_2 is the angle of the refracted ray entering at an angle θ' ($n_2 \sin\theta_2 = n_1 \sin\theta'$). To derive this equation from equation (12) in Ref. [6], we used the additional assumption that all distance scales in the image (including z_{int}) are small compared to the focal length of the lens. The corresponding detection PSF P_{det} is identical to P_{ilm} except for the removal of the $\sqrt{\cos\theta}$ and the replacement of k_{in} by the wave vector of the fluorescent light k_{out}. For an infinitesimal pinhole, the complete PSF is the product of these two point spread functions:

$$P(\mathbf{x}; z_{\text{int}}) = P_{\text{ilm}}(\mathbf{x}; z_{\text{int}}) P_{\text{det}}(\mathbf{x}; z_{\text{int}}) \quad . \tag{10}$$

The expressions in equations 9-10 are for a perfect pinhole confocal, whereas our confocal is a line-scanning confocal. While there have been several works describing line-scanning confocals [7, 8], these authors have treated where the line is focused onto the sample by a cylindrical lens. In our confocal, however, an image of a line is focused onto the sample through the large-aperture objective lens. As such, the illumination PSF in equation 10 is replaced by the integral of the detection PSF over a line in the x direction. These PSFs are illustrated in figure 4, showing the width of the PSF as a function of the index mismatch and the distance of the focus from the optical interface. The shift in the nominal focus position counteracts the slight decrease in the PSF's width close to interface for $n_2/n_1 < 1$.

We use this model for a line-scanning point spread function with aberrations as our model for our exact PSF, fitting the paramters that enter into equations 9-10. These parameters are the acceptance angle α of the objective lens, the wavelength of the laser, the ratio of energies of the fluorescent light to the excitation light, the index mismatch n_1/n_2 of the sample to the optics, the position of the optical interface z_{int}, and the amount that the lens is moved as the scan is rastered in z. In principle, other details could be included – polychromaticity and distribution of the fluorescent light, finite pinhole width of the illuminating line, etc. – but we find that these parameters are both relatively unconstrained by the fit and have little impact on the other reconstructed parameters, such as particle positions and radii.

In addition, for initial featuring we occasionally use a Gaussian approximation to the PSF. Based on calculations of the exact PSF, ≈ 90% of the function can be described by a Gaussian [4]. We verified this for PSFs calculated from Eq. 9, and found that although the presence of aberrations from the interface worsens the Gaussian approximation, generally a Gaussian accounts for ≈ 90% of the PSF except for in the most aberrated cases (large index mismatch imaging deep into the sample). Our simplest approximation of the PSF is as an anisotropic Gaussian with different widths in x, y, and z, with the widths changing with distance from the interface. In this simple approximation, we parameterize the Gaussian widths as a function of depth,

$$P(\mathbf{x}; z) = \prod_i \frac{e^{-x_i^2/2\sigma_i^2(z)}}{\sqrt{2\pi}\sigma_i(z)} \tag{11}$$

where each width $\sigma_i(z)$ is described by a polynomial in z, typically a second order Legendre polynomial.

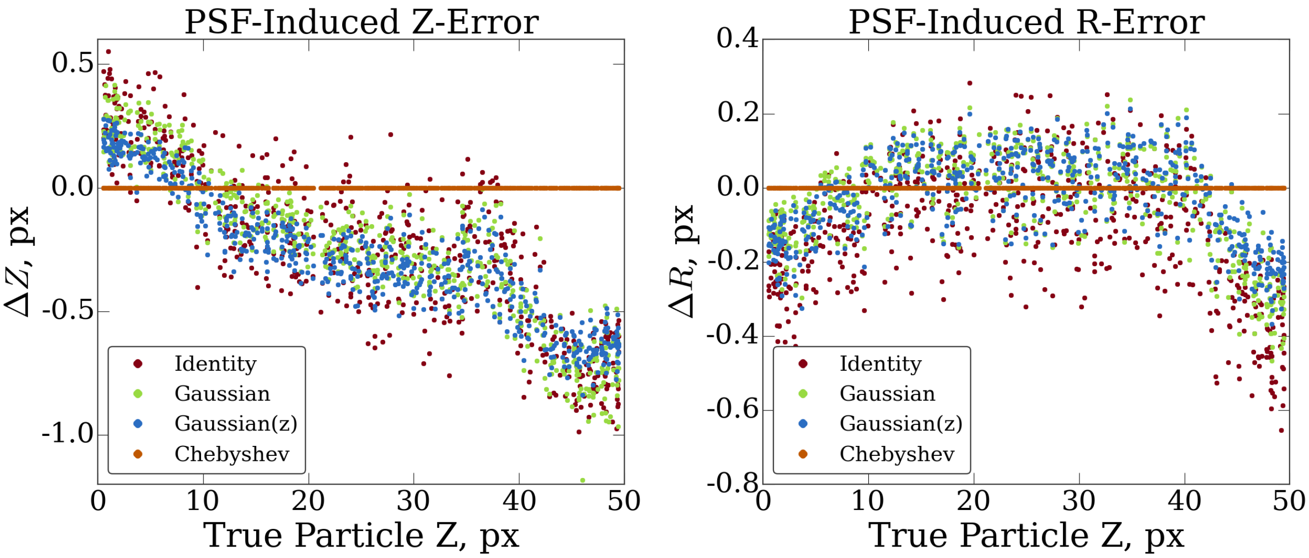

FIG. 5: **PSF generated biases**. Using an incorrect point-spread function results in significant biases, as PSF leakage affects neighboring particle fits. Moreover, since the PSF gets significantly broader with depth, using a spatially constant PSF, there are systematic biases with depth in both the z positions (left panel) and a characteristic drift in the fitted radii errors with depth (right panel), as shown for the delta-function (identity), an (x, y, z) anisotropic Gaussian, and a depth-varying Gaussian point-spread function. In contrast, using the correct Chebyshev PSF eliminates the errors in both the radii and z positions (data points forming thin orange line).

Figure 5 shows the effects of ignoring these details about the point-spread function on the extracted positions. We generate confocal images using a simulated, exact PSF with random distribution of particles up to a depth of 30 μm. Featuring these data using a 3D anisotropic Gaussian, we find a strong depth-dependent bias in the featured z position and radii measurements. Using a low order z-dependent Gaussian PSF decreases this bias only slightly. Interestingly however, ignoring the effects of diffraction completely and replacing the PSF with a Dirac delta-function does not cause significantly worse results than treating the PSF as a spatially-varying Gaussian. As shown by Fig. 5, an exact PSF is required to locate particle's positions and radii to within 20 nm (0.2 px). Therefore, we employ the full line-scan PSF calculation into our model.

The point-spread function defined in equations 9-10 decays extremely slowly with z and somewhat slowly in ρ. To accurately capture these long-tails of the PSF in our generative model, we calculate the PSF on a very large grid for convolutions, corresponding to ≈ $40 \times 25 \times 30$ px or ≈ $6 \times 3 \times 4$ μm in extent, which is considerably larger than the size of the 5 px radii particles. The long tails of the PSF bring information about structure far outside the image into the image region. As such, our generative model is defined not only in regions corresponding to the interior microscope image but also in an exterior padded region, which is cropped out when comparing to the model. For completeness, we still define the ILM and Platonic image (including exterior particles) in the exterior padded region; however parameters confined to this exterior region of the image are relatively unconstrained. We make up for this loss

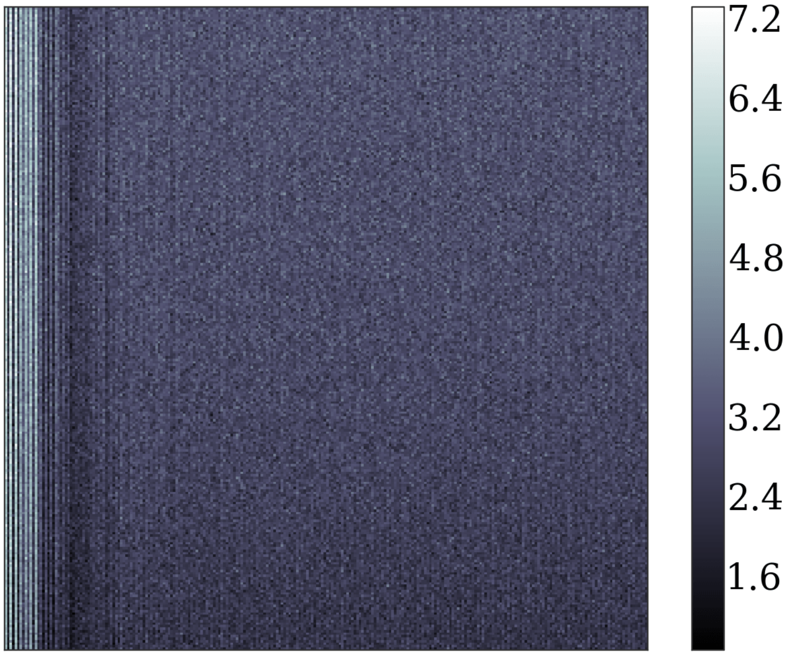

FIG. 6: **Experimental background image**. The measured background from our line-scan confocal microscope captured by adjusting the exposure to a full brightness image, removing the sample, and capturing a set of images with no illumination including room lights. Note that the range of values is from 1 to 7 out of a maximum 255 given by the 8-bit resolution of the CCD. While only a variation of 3%, we have seen in the illumination field section that this can create a bias that significantly alters our inference as a function of the position in the field of view. To remove this bias we fit the background field to a low order polynomial and add it to our model image.

in speed due to the increased size by doing an extremely accurate but approximate convolution based on Chebyshev interpolation, as described in a future paper.

D. Background

Due to background, the detector CCD pixels always read a non-zero value even when there is no light incident on them. We incorporate this into our generative model by fitting a nonzero background level to the images. Ideally, this background would be constant at every pixel location. Empirically, however, we find from blank images that this background varies with pixel location in the detector (see Fig.6). For our confocal microscope, we find the background is slowly-varying in the optical plane, perhaps due to different dwell times for different regions of the line scan and different sensitivies of different pixels; the background does not vary in z. As a result, the background is well-modeled by a low-order polynomial in x and y.

However, due to the long-tails of the PSF, the coverslip slab affects the image in a much larger z region than that of a typical particle. Rather than dealing with this by using an even larger point-spread function, we use the calculated point spread function to capture the effects of the PSF's moderate tails on the particles and slab, and fit a polynomial in z to capture the residual slab correction. This residual correction is mathematically the same as a background level in the detector. As a result, while the "true" background in the image is $P(x, y)$, our model uses a background $P(x, y) + P_{slab}(z)$, as the coverslip is usually oriented along the z direction.

E. Offset

In addition to a spatially-varying background field, we include an "offset" parameter c; this parameter captures rapid variations in the background. To demonstrate why we use an offset parameter, consider the model in equation 2. If there were no particles, $\Pi = 0$ and the model image would simply be $B(\mathbf{x}) + \int \mathrm{d}^3x\prime \; [I(\mathbf{x}')P(\mathbf{x} - \mathbf{x}'; \mathbf{x})$. Since B and I are arbitrary fields with no particles these two are degenerate in the fit space, regardless of the shape of the point-spread function P. When there are objects present and Π is not everywhere zero, this degeneracy is lifted. Here the model image will be $\mathcal{M}(\mathbf{x}) \approx I(\mathbf{x})$ in regions where there is dye, and $\mathcal{M}(\mathbf{x}) \approx B(\mathbf{x})$ where there are particles, if we ignore the effects of the point-spread function and do not include an offset ($c = 0$). Again, if this is all there were, this parameterization would be complete, and there would be no need for an offset c.

The use for an offset begins to appear when we settle on a fixed parameterization of $I(\mathbf{x})$ and $B(\mathbf{x})$. In our usage of PERI, we describe the background B with a fairly low-order polynomial, causing $B(\mathbf{x})$ to vary slowly with $\mathbf{x}$, and we describe $I(\mathbf{x})$ with a rapidly-varying field with many stripes. However, the real background in our confocal microscope does contain rapidly-varying structure. Including a constant offset c allows us to cheaply account for some of the rapid variations in B by coupling them to the rapid variations in I. For instance, consider the artificial case of an identity point-spread function. Then with an offset c equation 2 gives the model image as $\mathcal{M}(\mathbf{x}) = B(\mathbf{x}) + cI(\mathbf{x})$ in the regions where there are no particles, *i.e.* the offset includes some of the rapid variations of $I(\mathbf{x})$ in this region. Empirically, we find that including an offset parameter c significantly improves our reconstructed images and our extracted parameters. In addition, using an offset fitting parameter is considerably faster and more robust than attempting to reconstruct all of a spatially-varying background that is nearly degenerate with the illumination in much of the image.

$$\mathcal{M}(\mathbf{x}) = B(\mathbf{x}) + \int \mathrm{d}^3x' \; [I(\mathbf{x'})(1 - \Pi(\mathbf{x'}))] P(\mathbf{x} - \mathbf{x}'; \mathbf{x}) \tag{12}$$

F. Sensor noise

The last feature of the generative model is our understanding of the unrecoverable parts of the image: noise. To study the intrinsic noise spectrum of the confocal microscope, we subtract the long wavelength behavior from the blank image of Fig. 2. After removing the background we find that the noise appears white and is well approximated by a Gaussian distribution (see Fig. 7). There are, however, some highly localized non-Gaussian parts to the noise spectrum, arising due to the specific nature of our confocal. For instance, at high scan speeds slight intensity fluctuations in the laser's power couple to the dwell time on each stripe of line-scanned pixels. This produces periodic stripes across the image with a wavevector mostly parallel to the scan direction, but with a random noisy phase. How can we handle these sources of correlated noise and do they affect the quality of our reconstruction?

In principle, these correlated noise sources can be represented in the Bayesian model by introducing a full noise covariance matrix. That is, instead of writing that log-likelihood as the product of all pixel values, we can write

$$\log \mathcal{L}(\mathbf{M} \mid \mathbf{d}) = -\frac{1}{2}(M_i - d_i)\, \Lambda^{-1}_{ij}\, (M_j - d_j) \tag{13}$$

where Λ^{-1}_{ij} is the covariance matrix between each pixel residual in the entire image. In our optimization, we would form a low dimensional representation for this covariance matrix and allow it to vary until we find a maximum. In doing so, we would reconstruct the image and the correlated noise simultaneously. In practice, this introduces a large computational overhead due to the need for a full image convolution during each update as well as many new free parameters that need to be optimized.

Therefore, when desired we address the effect of correlated noise by working in reverse – we identify the several intense Fourier peaks in the confocal noise spectrum and remove them from the raw data before the fitting process. An example of this noise pole removal is given in Fig. 7. There, we can see that removing only 5 distinct poles (Fig. 7(d)) removes almost all visible correlated noise structure while changing the overall noise magnitude by a negligible amount. This small shift in estimated noise magnitude only affects the estimate of the errors associated with parameters such as positions and radii in a proportional way. Since these errors are very small and do not bias our inferred parameters, we often ignore the confocal's noise poles in our analysis entirely.

IV. MODEL CONSIDERATIONS

Here, we investigate several complexities of image formation in confocal microscopes and systematically analyze whether or not it is necessary to include them in our generative model. In particular, we will first analyze how much complexity we must introduce into the model elements listed in the previous section, including the platonic image, illumination field, and point spread function. We will also look at elements of image formation which we have not explicitly included in our model. First, confocal microscopes build a 3D image by rastering in 1, 2, or 3 dimensions (see section III). There is noise in this rastering procedure that affects the image formation process. Second, the

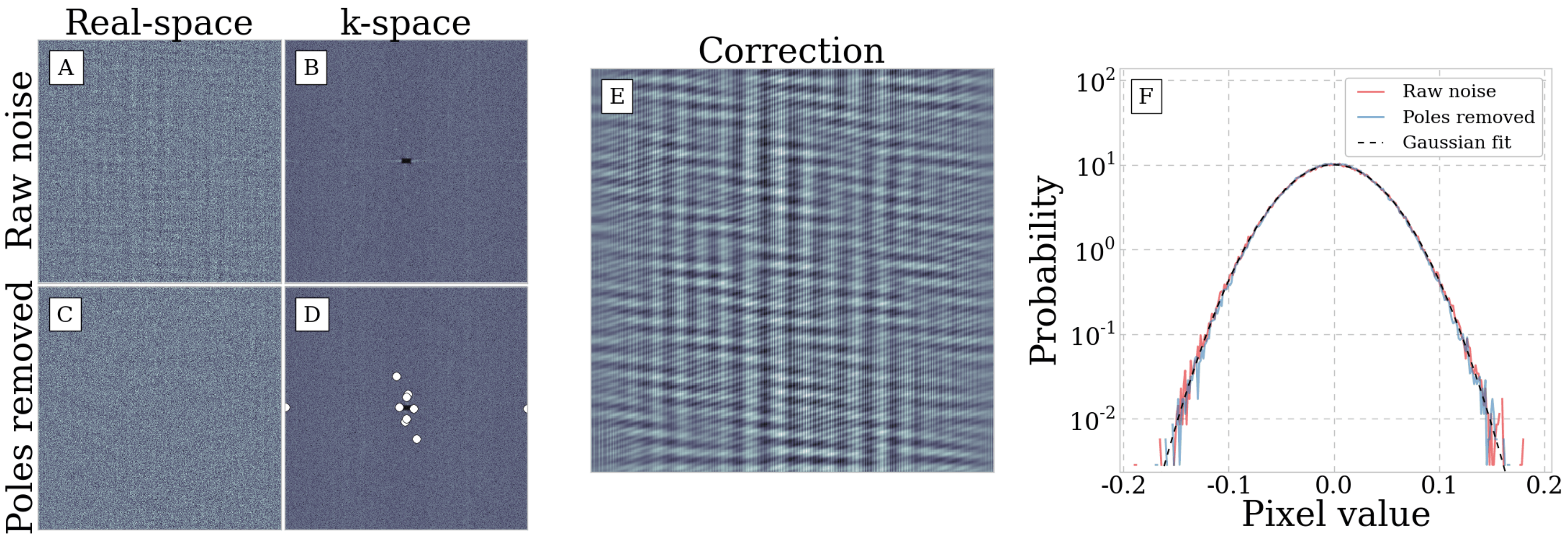

FIG. 7: **Noise spectrum** (a) Real-space plot of residuals representing the intrinsic noises generated in line-scanning confocal microscopy. This noise spectrum was generated by subtracting the background from a blank sample as in Fig.2. Notice that while most of the signal appears to be white noise, there is a systematic modulation along the x coordinate and high frequency features in the y scan direction. (b) Fourier power of the noise spectrum given in (a). The high frequency modulation can now be seen as two small 'poles' in the Fourier power spectrum. Note the dark box in the center of the spectrum is created by subtracting the high order polynomial background from the blank image. In (c) and (d) we present the real and Fourier space noise after removing several discrete peaks in the Fourier intensity that represent correlated noise sources. The removed signal can be seen in (e) showing the stripes created by the scanning nature of the confocal microscope. In (f) we show the histogram of residuals from (a) and (c). In solid red we plot the data and in dashed black lines we plot a Gaussian fit to the residuals with a width $\sigma = 0.0398$, showing that the noise spectrum is well approximated by a Gaussian distribution after taking into account long wavelength background features.

final image that comes from this scan is a cropped view of a much larger sample; the edges of this cropped image are influenced by the excluded exterior particles. Third, while the actual distribution of light intensity is a continuous field, the detector only measures a pixelated representation of this field. Fourth, while the exposure is made by the camera, particles undergo diffusional motion, blurring their apparent location. In this section, we address each of these image formation complexities and their effects on the inferred parameters.

We would like to systematically investigate at what level omitting a detail of the image formation from the model affects the fitted parameters. We can understand this quantitatively by examining the optimization procedure. Let us assume that the true image formation is completely described by a set of N parameters $\mathbf{\Theta}$. Then, near its maximum, the log-likelihood is approximately quadratic: $\log \mathcal{L} = \frac{1}{2}\sum_{ij} H_{ij}\Theta_i\Theta_j$, where the true value of the parameters is arbitrarily set to $\mathbf{\Theta} = 0$. Empirically, we find that with the starting parameter values provided by our initial featuring, the log-likelihood is extremely well-approximated by a quadratic.

If our model were complete, then the maximum of $\log \mathcal{L}$ would be exactly at the true parameter values $\mathbf{\Theta} = 0$. However, our model is incomplete. This means that, instead of fitting all N parameters $\mathbf{\Theta}$, we only fit the first (say) M parameters, which for convenience we denote as $\boldsymbol{\theta}$. Thus we can write the log-likelihood as three separate terms:

$$\log \mathcal{L} = \frac{1}{2}\sum_{i,j=1}^{M} H_{ij}\theta_i\theta_j + \sum_{i=M+1}^{N}\sum_{j=1}^{M} H_{ij}\Theta_i\theta_j + \frac{1}{2}\sum_{i,j=M+1}^{N} H_{ij}\Theta_i\Theta_j \quad . \tag{14}$$

The first term, containing only the parameters $\boldsymbol{\theta}$ that we are fitting, is the quadratic in the reduced space, with a maximum at the true parameter values. The unimportant third term reflects the separate contribution to $\log \mathcal{L}$ of the unknown or ignored portions of the model, and is constant in the $\boldsymbol{\theta}$ space. However, the second term mixes both the fitted parameters θ and the unknown parameters Θ_j. This mixing results in a linear shift of $\log \mathcal{L}$ in the $\boldsymbol{\theta}$ space away from the true parameters, and causes a systematic bias due to an incomplete model. Minimizing $\log \mathcal{L}$ with respect to θ gives the fitted values of the parameters gives an equation for the best-fit incomplete model parameters $\boldsymbol{\theta}$:

$$\theta_j = \sum_{k=1}^{M} \bar{H}^{-1}_{jk} \sum_{i=M+1}^{N} H_{ik}\Theta_i \tag{15}$$

where $\bar{H}^{-1}$ is the inverse of the sub-block $\bar{H}$ of the Hessian matrix H that corresponds to the fitted parameters $\boldsymbol{\theta}$.

We can use equation 15 to estimate the effect on one of the estimated parameters θ_j, if we ignore one aspect of the generative model Θ_k. Ignoring the off-diagonal terms in H^{-1} to capture the scaling gives $\theta_j \approx H_{kj}\Theta_k/H_{jj}$. Thus, the error in the fitted parameter θ_j is proportional to both the coupling H_{kj} between that parameter and the ignored aspect of the generative model, and the magnitude of the error of the generative model Θ_k.

A. Component complexities

There are several choices one can make concerning the form and complexity of each of the components of our model image. As discussed in the Section III, we have implemented many forms of the platonic image, illumination field, and point spread function and each one of these forms has a varying number of parameters with which to fit. How do we decide which form to use and at which complexity (number of parameters) to stop? To decide on a per-image basis, we could employ Occam's factor, which is a measure of the evidence that a model is correct given the data [9]. In practice, however, we are mainly concerned with how these models influence the underlying observables which we are attempting to extract. That is, we wish to use knowledge of the physical system to check which model best predicts the particle locations and sizes. To do so (as mentioned in the main manuscript), we often turn to particle sizes versus time as well as particle overlaps, both physical statements that assert almost no assumptions on our system.

We can also get a sense of the magnitude of the effect these choices have on inferred positions and radii by creating synthetic data and fitting it using a simpler model. In Fig. 8 we show the residuals of such fits for various simplifications made to the platonic form, illumination field, and point spread function. In the left columns of the figure we see the reference image formed using the most complex image model available and in each row the residuals for each choice with a description of that choice above the panel. For all but the last column, in which we fit the image with the exact model once again, we can see systematic errors in the fit. We compute how much these residuals influence the extracted positions and radii and report these errors in Table I. In particular, most choices of platonic image aside from the naive boolean cut do not influence particle featuring below an SNR of 30. However, the complexity of the illumination field always matters until all long wavelength structure is removed from the image. Finally, the choice of PSF is crucial, requiring the use of a calculated confocal PSF to even approach the CRB.

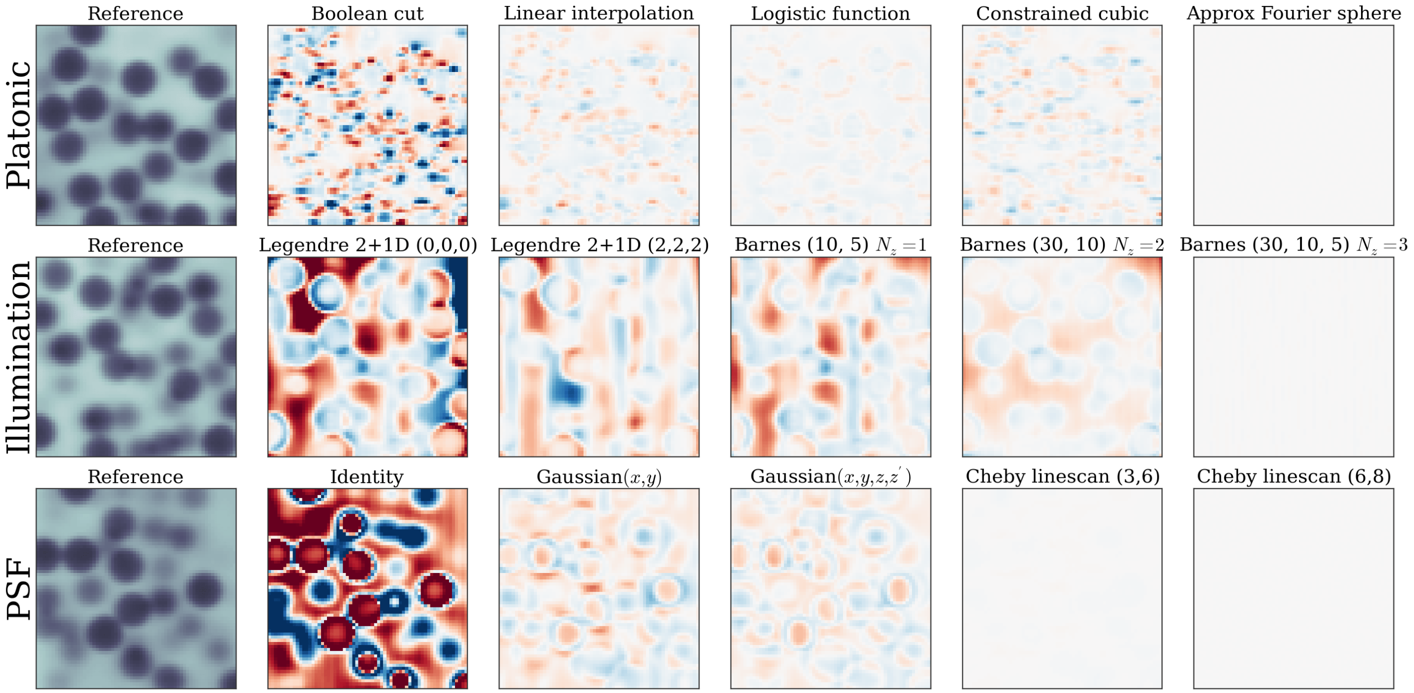

FIG. 8: **Component complexity residuals**. Here we visually demonstrate the results of choosing between different forms of model components as well as different parameterizations of a single component. We generate simulated microscope images using the model components which we employ when fitting experimental data (left column) and fit them with different choices of platonic image (top row), illumination field (middle row), and point spread function (bottom row). Each choice is labeled above its panel showing the residuals and each row is on a common color scale. In the case of the platonic forms, the boolean cut, linear interpolation, and constrained cubic display higher order multipole errors while the logistic function's first correction is to the monopole (volume) term (as shown by the presence of rings). In the illumination field, stripes are present in the residuals until we use a Barnes interpolant with 30 control points. Past that, the ability to capture intensity as a function of depth is the remaining term which we are able to fit with a single extra Legendre polynomial in the z-direction. Finally, in the case of the PSF, we see hard boundaries transitioning to softer boundaries using a Gaussian PSF in both 3D and 3+1D. The residuals all but disappear when the image is fit with our exact line-scan confocal PSF model (Eq. 9) approximated by a Chebyshev polynomial in 3+1D.

	Fitting model type	Position error (px)	Radius error (px)
Platonic form	Boolean cut	0.03376	0.01577
	Linear interpolation	0.00778	0.00386
	Logistic function	0.00411	0.00352
	Constrained cubic	0.00674	0.00249
	Approx Fourier sphere	0.00000	0.00000
Illumination	Legendre 2+1D (0,0,0)	0.18051	0.13011
	Legendre 2+1D (2,2,2)	0.06653	0.03048
	Barnes (10, 5) $N_z = 1$	0.13056	0.06997
	Barnes (30, 10) $N_z = 2$	0.04256	0.02230
	Barnes (30, 10, 5) $N_z = 3$	0.00074	0.00022
PSF	Identity	0.54427	0.57199
	Gaussian(x, y)	0.47371	0.14463
	Gaussian(x, y, z, z')	0.34448	0.04327
	Cheby linescan (3,6)	0.03081	0.00729
	Cheby linescan (6,8)	0.00000	0.00000

TABLE I: **Position and radii errors by model complexity.** Here we tabulate the position and radius errors associated with the model component choices made in Fig. 8. Note that while the components with the largest impact on determining underlying parameters are the ILM and PSF, the choice of platonic image cannot be ignored in order to reach the theoretical maximum resolution. Interestingly, in the case of PSF selection, Gaussian(x, y, z, z') (3+1D) is almost no better at extracting particle positions than Gaussian(x, y) (2D). However, its ability to extract particle sizes increases by 3 since it takes into account the variation of the PSF in space. Additionally, in the case of the ILM, capturing the stripes in the illumination using a 30 control point Barnes increases the resolution by 3 whereas capturing the illumination's dependence in depth causes the resolution to increase 10 fold.

B. Scan jitter

Confocal microscopes operate by taking an image with the lens at a fixed z position to create one layer of the three-dimensional image, then moving the lens up a fixed amount to take the next layer. In our generative model, we assume that these steps of the lens (and the resultant image slices) are perfectly equally spaced by an amount which is fitted internally. However, a real confocal microscope will have some error in the vertical positioning of the lens. As a result, the actual image taken will not be sampled at exactly evenly spaced slices in z, but at slices that are slightly shifted by a random amount.

To test the effect of this z-scan jitter on our parameter estimation, we simulate images taken by a confocal microscope with imperfect z-positioning. Instead of sampling the image at a deterministic z position, we instead sampled the image at a z position shifted from the ideal position by an uncorrelated Gaussian amount of varying standard deviation. A representative image of a 5 px radius particle with a step positioning error of 10% is shown in Fig. 9(a). There is very little difference between this image with z jitter and the perfectly-sampled image, as shown by the difference image in panel b. We then fit an ensemble of these images at varying image SNR levels, over a random sampling of image noise, z-jitter noise, and random shifts of particle positions by a fraction of a pixel.

The results of these fits are shown in Fig. 9c, showing the actual error in the featured positions versus the size of the z-positioning noise. For our confocal which is equipped with a hyper-fine z-positioning piezo, we expect the z positioning error to be a few nm, or a few percent of a pixel. For a 3% error in positioning, the signal-to-noise ratio must be ≈ 100 for the effects of z-positioning jitter to be comparable to the theoretical minimum effect from the image noise. This small effect of the error is partially due to the large size of our particle. If each z slice of the image is randomly displaced with standard deviation σ, then we expect roughly a $\sigma/\sqrt{N}$ scaling for the final error in the particle's z-position, where N is the number of z slices the particle appears in. A 5 px diameter particle with a 4 px axial point-spread function occupies ≈ 18 difference slices, decreasing the effect of scan noise by a factor of ≈ 4 and putting it below the CRB for our data.

As the error in z-positioning increases, however, the effect on the featured particle positions increases correspondingly. The error due to a $\approx 10\%$ z jitter is comparable to the CRB for image noises of $SNR = 20$. For exceptionally large z-jitters of 40% the error due to the lens positioning dominates all other sources of error. However, even with this large error in lens positioning, the error in featured positions is still only 10% of a pixel, or about 10 nm in physical units.

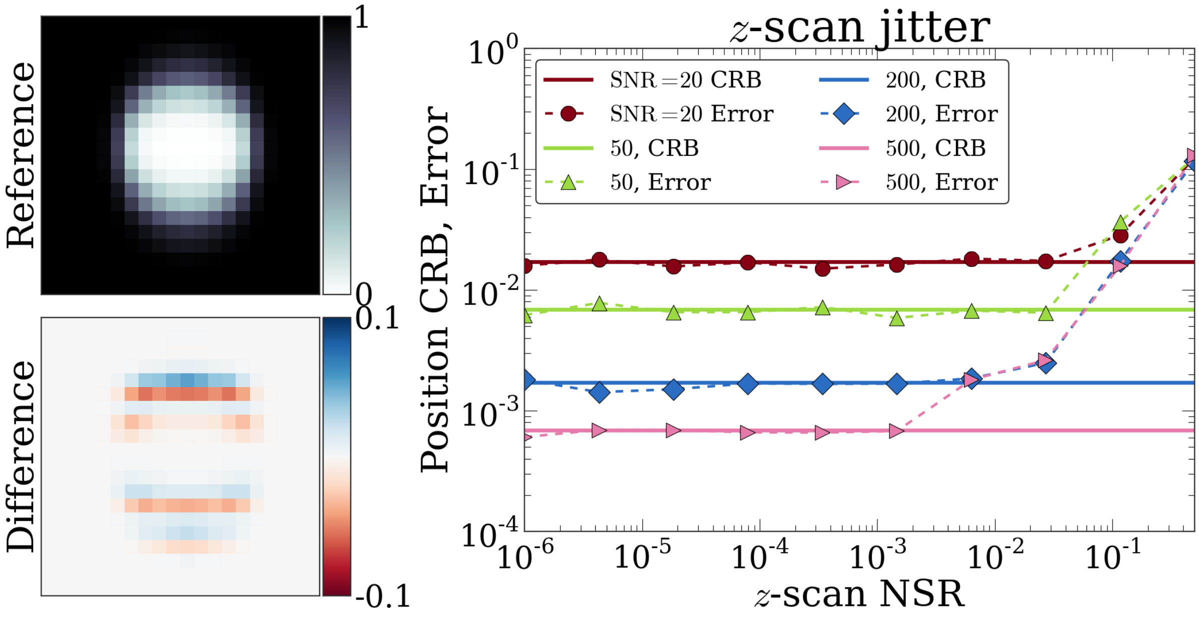

FIG. 9: **Lens Positioning Jitter** (a) The xz cross-section of a simulated image of a 5 px radius colloidal particle taken with a 10% error in the lens positioning. (b) The difference between the image with positioning error and a reference image with zero positioning error. The differences between the images are both random and small, for this image no more than 7% of the perfect image intensity. (c) The effect of lens positioning error on featured particle positions, at signal-to-noise ratios of 20, 50, 200, and 500. The solid symbols and dashed lines show the position error for images with imperfect lens positioning, while the solid lines denote the Cramer-Rao bound for an image with no positioning error. At lens positioning errors of ≈ 10% or larger, the error in featured positions from the z-slice jitter dominates that from the simple image noise, even for an SNR of 20. However, the featuring error due to a z jitter of ≈ 1% is less than the error due to image noise, for any noise level than can be captured by an 8-bit camera.

C. Missing and Edge particles

The point spread function delocalizes the particle's image over a region larger than the particle's size. As a result, if two particles are close enough together, their images can overlap. This overlapping is a significant problem for heuristics such as centroid fitting, as the true particle centers do not coincide with the fitted centroid. In contrast, PERI's accuracy is negligibly affected by the presence of a second, close particle, since PERI correctly incorporates close particles in its generative model. The CRB of two touching, 5 px diameter particles increases by only ≈ 3%, and PERI finds particles to this accuracy when close.

However, large systematic errors can affect PERI when one of these particles is missing in the generative model. This situation is illustrated in its simplest form in Fig. 10. If one of the two touching particles is missing from the generative model, then the second particle will be enlarged and drawn into the first particle's void to compensate, as shown in panel b. As a result, the missing second particle will severely bias the fitted positions and radii of the first particle. Fig. 10c shows the magnitude of this effect. For particles separated by 1 px or less, significant biases on the order of 0.4 px appear in the identified particle's featured position. These biases matter at essentially all values of the SNR, only being comparable to the CRB for SNR < 1. As a result, it is essential for PERI to identify all the particles in the image to return accurate results. For this reason, we take extra precaution and thoroughly search the image for missing particles before fitting, as detailed in section V.

The biases caused by missing particles appear whether or not the missing particle is located inside or outside the image. As a result, accurately locating edge particles requires identifying all their nearby particles, even ones that are outside the image! We attempt to solve this problem by padding the Platonic and model images and the ILM by a significant portion, and including this padded extra-image region in both the add/remove and relaxation portions of the PERI algorithm. Nevertheless, it is extremely difficult to locate all the particles outside the image, for obvious reasons. As such, there is the possibility for moderate systematic errors to enter for particles located at or near the edge.

Nevertheless, if the exterior particle is identified, PERI correctly locates the interior particle, as shown in Fig. 11. To demonstrate this, we create simulated images of two particles near the boundary of an image. One particle is

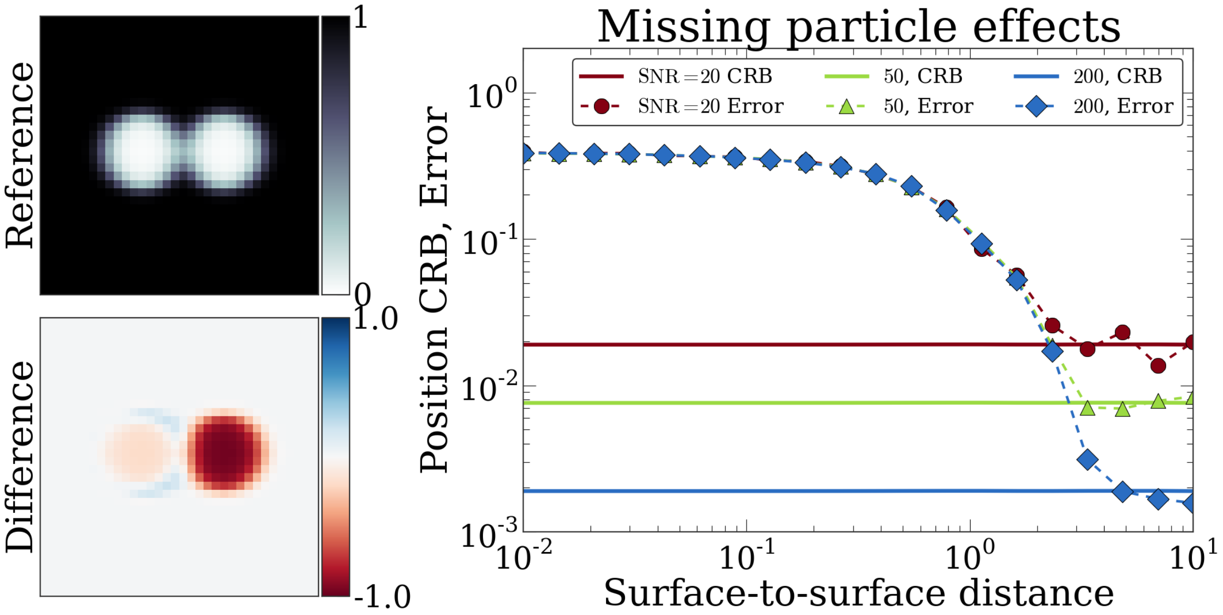

FIG. 10: **Effect of missing particles**. (a) The xz-cross section of an image of two 5 px radius particles placed in contact. (b) The difference image for a bad generative model that includes only the particle on the left. To minimize the effect of the missing right particle, the left particle is drawn to the right and expanded in radius. This effect is visible as the red and blue ring on the right border of the left particle. (c) The error in position along the separation axis, as a function of true surface-to-surface distance, for a model with a missing particle. When the particles are separated by $\approx$ 10 px the featured particle is located correctly. However, as the particles get closer than $\approx$ 2 px significant biases start to appear. These biases saturate at a separation of $\approx$ 0.1 px, corresponding to a featuring error of $\approx$ 0.4 px.

placed at $z = a$ so that its edge just touches the boundary while the other is placed at $z = -(a + \delta)$ on the other side of the border. We plot the CRB of the interior particle and the measurement errors of both PERI and trackpy [10] as a function of the exterior particle's coordinate in Fig. 11. While the CRB only changes by a factor of 2 as the particles come within contact, the featuring errors grow drastically for traditional featuring methods due to biases introduced by the exterior particle. For this same data set, PERI featuring errors follow the CRB allowing precise unbiased featuring of particles at the edge of images.

This apparent conundrum of edge particles presents an interesting positive side-effect. Missing edge particles affect the fits because they contribute a significant amount to the image. As such, we might expect that a particle outside the field of view can still be located very precisely. This prediction is borne out by a calculation of the Cramér-Rao bound, as shown in Fig. 12. Until the particle and PSF fall off the edge of the image (distance $> 1R$), the CRB remains constant for all particle parameters. When the particle is centered on the image edge (distance of 0), the CRB is twice that of the bulk, intuitively corresponding to a loss of half of the information about the particle. As the volume of the particle leaves the image, the CRB decreases as $1/\delta^2$ until the particle is no longer part of the image. Interestingly, Fig. 12 shows that the PSF constrains the particle position to within 0.1 px even when the particle is entirely out of the image! If correctly seeded with a moderate guess for the particle position outside the image, PERI will locate the particle to a precision of the Cramér-Rao bound. However, in practice it is very difficult to seed these particles into PERI, as a slight change of the intensity at the image edge could be either a missing particle outside the image or a slight variation in the ILM near the image edge. Nevertheless, PERI is very good at locating particles that are partially outside the image.

D. Pixel intensity integration

Our generative model considers the image formed on the camera as if the camera pixels had an infinitesimal size. In reality, the camera pixels have a finite extent. As a result, the image at each pixel on the camera is not a discrete sampling of the light intensity, as in our generative model, but is instead an integration in the detector plane over the pixel's size.

To check whether the effect of pixel integration matters, we generated images that were up-sampled by a factor of

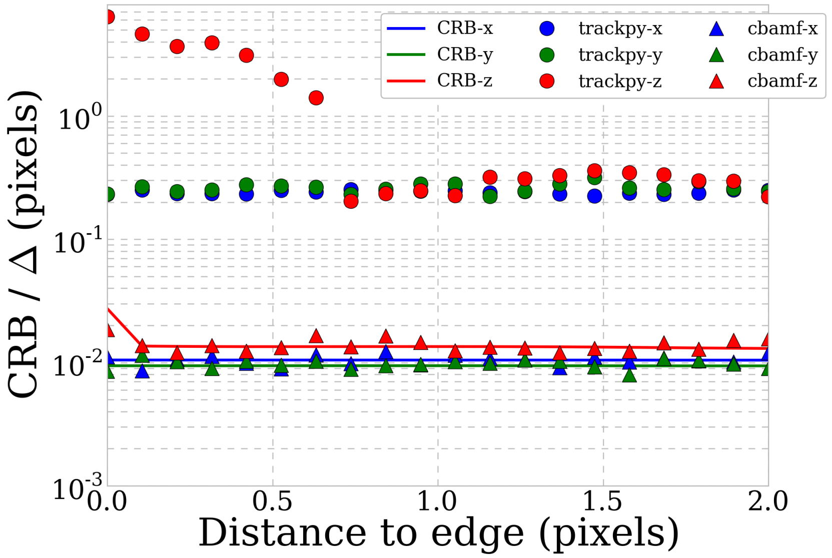

FIG. 11: **Influence of particles outside of the image**. Here we place one particle at $x = a$ and a second particle at $x = -(a + \delta)$ so that one is completely inside the image and the other outside. We plot the CRB for the x, y, and z positions and radius a of the interior particle as well as measured errors for PERI in triangles and a centroid algorithm (trackpy [10]) in circles as a function of the position of the second particle. When the exterior particle is further than a pixel outside the image we see that the measurements of the interior particle are constant. However, as the PSF of the exterior particle begins to overlap the interior particle the CRB and all measured errors increase dramatically. While PERI's measured error continues to follow the CRB, trackpy's error increases beyond pixel resolution. Note that pixel separations at the edge are generic in colloidal images especially in dense suspensions.

8 in the xy-plane. We then numerically integrated these images over the size of each pixel. A representative image is shown in Fig. 13a. There is very little difference between the xy-integrated image and the generative model, as visible in panel b. We then fitted an ensemble of these xy-integrated pixel images, both over an ensemble of noise samples and over an ensemble of particle positions shifted by a random fraction of a pixel. The results are shown in Fig. 13c. We find that there is no discernible effect of pixel integration at a SNR of 200 or less. The error due to neglecting pixel integration becomes comparable to that due to noise only for $SNR \geq 400$, which is significantly higher than the maximum allowed by an ordinary 8-bit camera. Thus, the effect of integrating over a pixel size for a colloidal particle essentially always has a negligible effect on the fitted parameters.

E. Diffusional motion

A typical colloidal particle is not fixed in its location, but diffuses about due to Brownian motion. For an isolated colloidal particle, this Brownian motion results in a random walk with mean displacement $\langle \mathbf{x} \rangle = \mathbf{0}$ and a mean-square displacement $\langle x^2 \rangle = 6Dt$ that is linear in time, with a diffusion constant $D = kT/6\pi\eta R$ where η is the solvent viscosity and R the particle radius. As a result, the microscope takes an image not of a colloidal particle at a single position, but of an integrated image of the colloidal particle over the trajectory that it has diffused.

First, at what length- and time- scales is a colloidal particle de-localized due to Brownian motion by a scale that is larger than the resolution? For a 1 μm diameter particle in water to diffuse the 1 nm resolution provided by PERI takes a fantastically small time of $t = 1$ nm^2/D $\approx 10\mu$s. Even for our relatively viscous samples of $\approx 80\%$ glycerol and 20% water this time slows down to only $\approx 600\mu$s. These times are orders of magnitude faster than the $\approx$ 5ms required by our confocal to take a 3D image of the particle, corresponding to a 8 nm displacement. Thus, a freely diffusing particle has always diffused much further than the positional featuring uncertainty intrinsic to PERI.

However, this does not mean that the precision past 8 nm is empty. The particle's positions are Gaussian distributed about its mean value during the exposure time. While the extent of the distribution is much larger than the PERI featuring errors, the particle's mean position during the exposure time is well-defined. Moreover, the actual image on the camera from the diffusing particle is a convolution of the particle's trajectory with a single particle image. Since

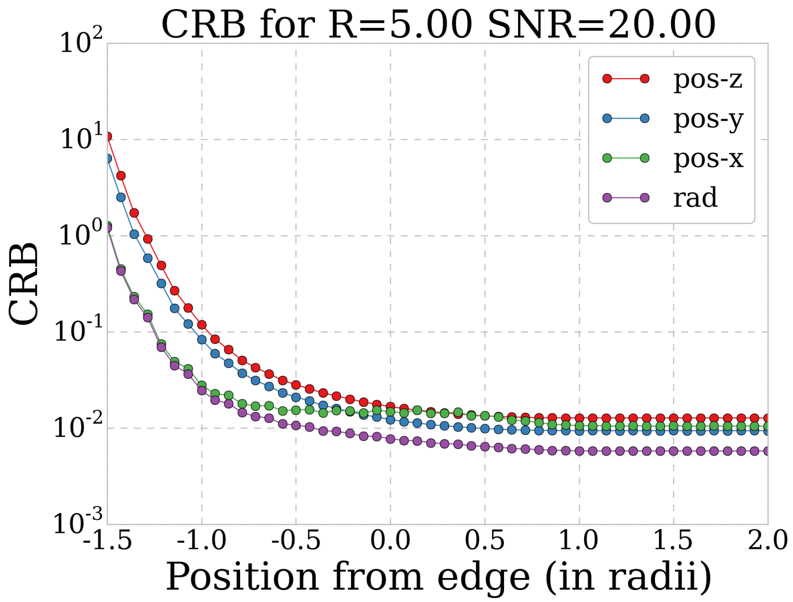

FIG. 12: **CRB of edge particles**. Here we calculate the Cramér-Rao bound of the x, y, and z positions as well as radius (in red, blue, green, purple respectively) for an isolated particle as a function of its distance to the edge of the image. For positive displacement (inside the image) we see very little change with position as expected. As parts of the PSF leak out of the image (displacements close to zero, positive) we see that the expected error increases slightly since information is lost. Finally, as the particle itself leaves the image, information is lost more dramatically as indicated by a sharp rise in the CRB. However, note that even at a displacement of one radius a, the PSF allows us to locate the particle outside of the image to within a pixel. While in practice it is difficult to identify these particles systematically, their presence can greatly influence the measured positions of other edge particles.

this convolution is like an averaging, we might expect that the small Brownian excursions are averaged out in the image formation, and that the image allows for accurate featuring of the particle's mean position.

We can use the formalism of Eq. 15 to show that Brownian motion does not affect our featuring accuracies. Let the particle's mean position be $\bar{\mathbf{x}}_0$, and its Brownian trajectory be $\mathbf{x}_0(t)$. Then the actual image $I(\mathbf{x})$on the detector is

$$I(\mathbf{x}) = \frac{1}{t_{\text{exp}}} \int_0^{t_{\text{exp}}} I_0(\mathbf{x}_0(t))\, dt = I_0(\bar{\mathbf{x}}_0) + \frac{1}{t_{\text{exp}}} \int_0^{t_{\text{exp}}} I_0(\mathbf{x}_0(t)) - I_0(\bar{\mathbf{x}}_0)\, dt \tag{16}$$

where $I_0(\mathbf{x})$ is the image of one particle at position $\mathbf{x}$ and t_{exp} is the camera exposure time. We view the actual image as $I(\mathbf{x}) = I_0(\bar{\mathbf{x}}_0; \theta) + (1-\Theta)\Delta I$, in terms of a group of fitted parameters $\boldsymbol{\theta}$ and an additional parameter Θ describing the effects of Brownian motion ΔI. For the true image $\Theta = 0$ but for our model image $\Theta = 1$. Then equation 15 says the error will be $\theta_j \approx H_{kj}/H_{jj}$, where $H_{\Theta j} = \partial_\Theta \partial_{\theta_j} I = \partial_{\theta_j} \Delta I$. However, for small displacements the effect of Brownian motion on the image is

$$\Delta I = \frac{1}{t_{\text{exp}}} \int_0^{t_{\text{exp}}} \frac{\partial I(\bar{\mathbf{x}}_0)}{\partial x_i} (\mathbf{x} - \bar{\mathbf{x}}_0)\, dt = 0$$

since $\partial I(\bar{\mathbf{x}}_0)/\partial x_i$ does not depend on time. As a result, $\partial_{\theta_k} \partial_\Theta I = 0$ and there is no effect of Brownian motion on the image to first order in the displacements, *i.e.* when the particle displacement is moderately small compared to the radius.

Finally, in Fig. 14 we show empirically that the effect of Brownian motion is negligible for our exposure times. To create an image of a diffusing particle captured by a slow camera, we simulated a 200 point Brownian trajectory of a $R = 5$ px radius particle, generating an image for each point in the particle's trajectory. We then took the average of these images as the noise-free image captured by the microscope. One such image is shown in Fig. 14a. Once again, there is a slight difference (10%, as shown in panel b) between the slow image of a diffusing particle and the reference image taken of a particle at a single location. We then fitted an ensemble of these images, over a variety of both Brownian trajectories and noise samples. Fig. 14c shows the results of these fits as a function of the mean displacement during the collection $\tau_{\text{exposure}}/(R^2 D)$, where τ_{exposure} is the exposure time of the camera and D the particle's diffusion

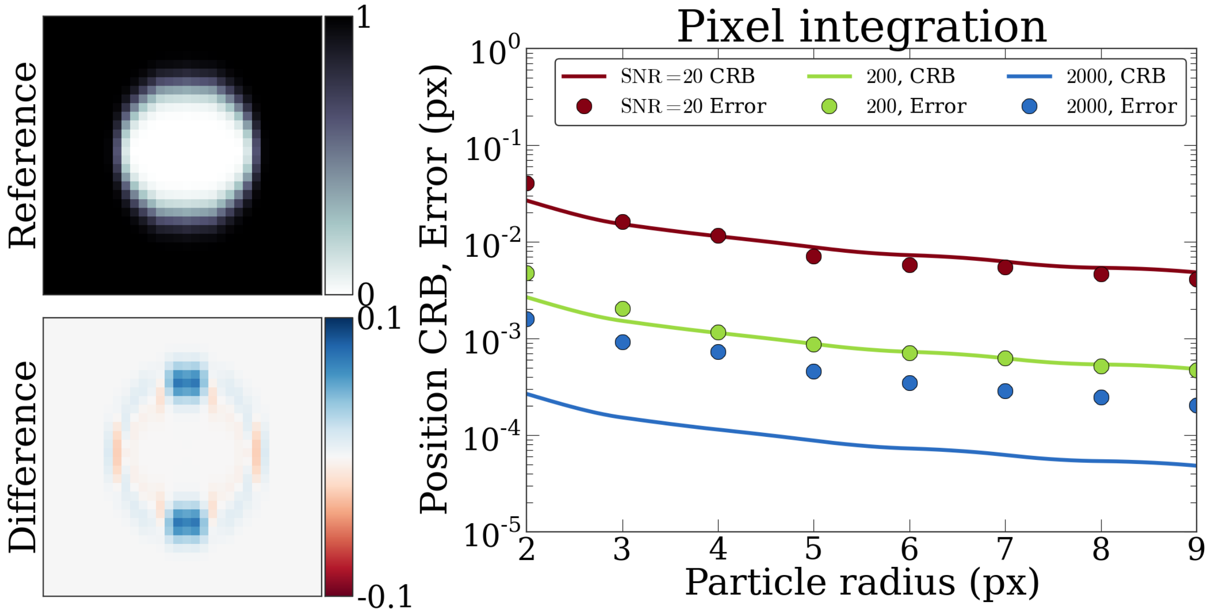

FIG. 13: **Pixel Integration** (a) The xz cross-section of a simulated image of a 5 px radius colloidal particle, where each pixel contains the light intensity integrated over its area instead of sampled at its center. (b) The difference between the pixel-integrated image and a reference image sampled at the center of the pixels. The differences between the images are small (10%) and centered in a ring which has mean 0 and is positioned at the particle's edge. (c) The effect of pixel integration on featured particle positions as a function of particle radius, at signal-to-noise ratios of 20, 200, and 2000. The solid symbols and dashed lines show the position error for images generated with pixel integration and fit without, while the solid lines denote the Cramer-Rao bound for the images (without pixel integration). Integrating over a pixel area has no effect on the featured positions for any SNR compatible with an 8-bit depth camera. The effect of pixel integration only starts to matter for an SNR $\geq$ 400 (not shown).

constant. Brownian motion has a negligible effect on the featured positions for our experimental images of freely-diffusing particles (camera exposure time of 100 ms and $D = 0.007\ \mu\text{m}^2/\text{s}$ corresponding to a 1 μm particle in 80:20 glycerol:water, corresponding to $\tau_{\text{exposure}}/(R^2 D) \approx 10^{-3}$). Interestingly, however, to achieve a higher localization accuracy at a higher SNR of ≈ 200, Brownian motion must be correctly taken into account in the image formation. Incorporating Brownian motion at these high signal-to-noise ratios would allow the teasing out of information about the particle's trajectory from a single image.

V. IMPLEMENTATION

A typical confocal image is roughly 512 x 512 x 100 pixels in size and contains 10^4 particles meaning that the number of degrees of freedom in our fit is roughly 10^7 described by 10^5 parameters, a daunting space to optimize. On modern hardware using the highly optimized FFTW, the typical time for an FFT the size of a single image is $\sim$ 1 sec. Given this time, a single sweep through all parameters would take an entire week while a full optimization would consume a year of computer time. However, since particles have finite size, we are able to optimize most of these parameters locally with a small coupling to global parameters (ILM, PSF). Additionally, the finite intensity resolution of microscope sensors, typically 8 or 16 bits, allows us to make further simplifications to our model. Here we describe the practical algorithmic optimizations that we have made as well as the optimization schedule that we have devised to quickly reach the best fit model.

A. Partial image updates

First and foremost, we optimize our fitting procedure by working in image updates and only updating parts of the image that are required at any one time. In order to modify the position of one particle by a small amount, the number of pixels that are affected is simply $(2a + w)^3$ where a is the particle radius and w is the PSF width, both in

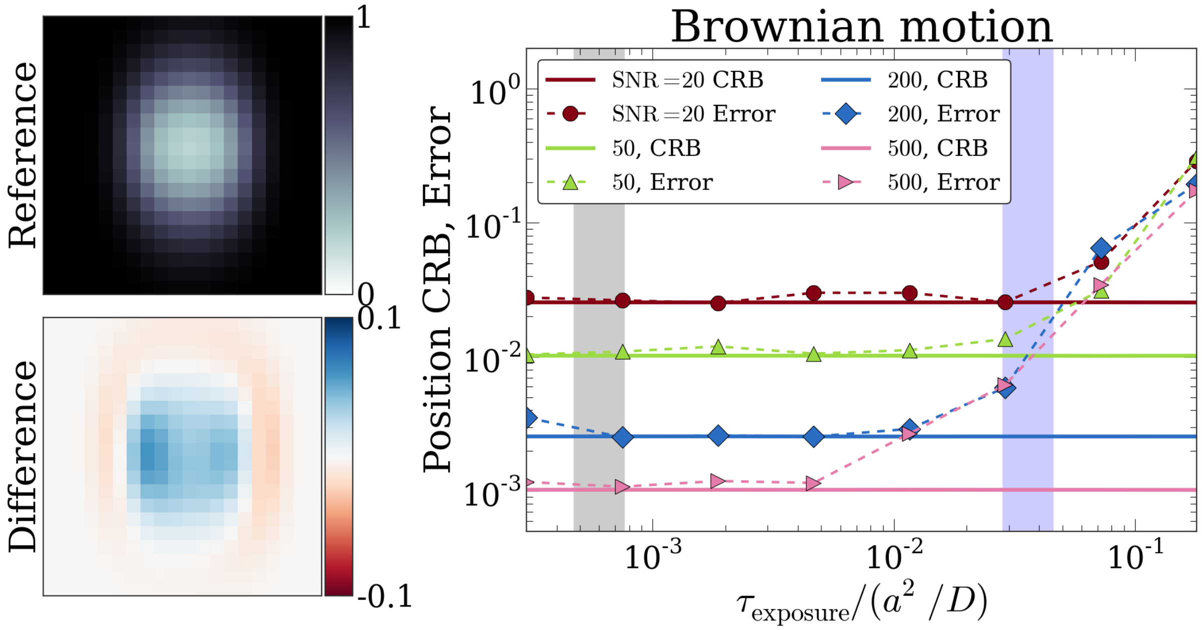

FIG. 14: **Brownian Motion** (a) The xz cross-section of a simulated image of a 5 px radius colloidal particle undergoing strong Brownian motion $\tau_{\rm exposure}/(R^2D) = 0.01$ during the image formation. (b) The difference between the diffusing-particle image and a reference image without diffusion. The differences between the images are small (10%) and are mostly in a ring with mean 0 at the particle's edge. (c) The effect of Brownian motion on featured particle positions as a function of exposure time, at signal-to-noise ratios of 20, 50, 200, and 500. The image exposure time for our camera is located in the shaded grey band for 20/80 water/glycerol and blue band for pure water. The solid symbols and dashed lines show the error between the fitted positions and the mean position in the particle's trajectory, while the solid lines denote the Cramer-Rao bound for the generated images. At our exposure times and SNR of 20, the effects of Brownian motion are small compared to those from noise in the image. Interestingly, for higher SNR or slower exposure times, Brownian motion starts to have a noticeable effect and must be incorporated into the image generation model.

pixels. For a typical particle, the ratio of this volume to the entire image volume is typically 10^{-2} which represents a speed up of the same factor due to the roughly linear scaling of FFT performance with problem size ($N \log N$). In addition, since the PSF decreases with distance from a particle's center, a localized object produces only a weak signal in regions far away from it. For confocal microscope PSFs, the distance scale associated with this signal change is only a few tens of pixels. Therefore, we employ a technique common applied to inter-atomic potentials in molecular dynamic simulation – we simply cutoff the PSF at this distance scale allowing for exact partial updates. By cutting off the PSF, we are able to incrementally apply image updates in an exact procedure (up to floating point errors). For example, when moving a single particle from $\mathbf{x}_0$ to $\mathbf{x}_1$, we must simply calculate the local image change given by

$$\Delta\mathcal{M}(\mathbf{x}) = \int \mathrm{d}^3x' \; [I(\mathbf{x'})(1-c)(\Pi(\mathbf{x};\mathbf{x}_1) - \Pi(\mathbf{x};\mathbf{x}_0))]P(\mathbf{x}-\mathbf{x}';\mathbf{x}) \quad , \tag{17}$$

cf. equation 2, then calculate $\mathcal{M} + \Delta\mathcal{M}$ only in a small local region around the particle being updated. We are able to use similar update rules for all variables except for those effecting the entire image such as the PSF, offset, $z_{\rm scale}$, and estimate of the SNR.

Additionally, in our code, we generously employ the principle of "space-time trade-off" in which we cache intermediate results of all model components and reuse them later in the computation. In particular, we maintain a full platonic image and illumination field, which we update along with the model image. We also cache the calculated PSF so that we may utilize previous results until the PSF is sampled. In doing so, we are limited in our current implementation by the speed of the FFT, which takes 70% of the total runtime.

B. Optimization of parameters and sampling for error

Once an approximate initial guess is obtained by more traditional featuring methods [11], we optimize the parameters by fitting using a modified Levenberg-Marquardt routine. Our Levenberg-Marquardt algorithm uses previouslyreported optimization strategies designed for large parameter spaces [12]. However, a Levenberg-Marquardt minimization requires the matrix $J_{i\alpha} \equiv \partial m(x_i)/\partial\theta_\alpha$, which is the gradient of each pixel in the model with respect to all the parameters. For the $\approx 10^5$ parameters and 10^7 pixels in our image, this matrix would be many thousand times too large to store in memory. Instead, we construct a random approximation to $J_{i\alpha}$ by using a random sub-section of pixels x_i in the image to compute J. This approach works well for the global parameters (PSF, ILM, etc) but fails for the particles, which appear in a relatively small number of pixels. For the particles, we instead fit small groups of adjacent particles using the full $J_{i\alpha}$ for the local region of affected pixels. As the global parameters and particle parameters are coupled, we iterate by optimizing first the globals, then the particles, and repeating until the optimization has converged.

Once the model is optimized, we can employ two different methods to extract the errors associated with each parameter. Since we calculated the gradients J during the optimization procedure, we can use this to find the covariance matrix $(J^T J)^{-1}$ which gives the correlated sensitivities of each parameter. In practice this is the faster method and yields accurate results and, as such, is our method of choice. However, additionally, we may use Monte Carlo sampling to estimate parameter errors. Our Monte Carlo sampler sweeps over each parameter and updates the particle position, accepting or rejecting based on the change in the log-likelihood of the model. We use slice sampling to produce highly uncorrelated samples, allowing an excellent error estimate from only a few sweeps. Our error sampling doubles as a check for convergence. If the log-likelihood increases after sampling, then the optimization has not converged and either more Monte Carlo sampling or more traditional optimization is needed. In practice, when desired we check with $\approx 5-10$ Monte Carlo sweeps, and ensure that the log-likelihood remains the same or fluctuates by a few times $\sqrt{N}$, where N is the number of parameters in the model.

C. Identifying missing and mis-featured particles

Fitting the entire image allows us to easily identify any missing or mis-identified particles. To do this, we first start with an initial guess for the particle positions from a standard centroiding algorithm, and we fit those particle parameters and global microscope parameters moderately well, using the procedure described above. (We do not fit the image completely here to save time.) At this point, regions of the image with all particles correctly identified are fairly well-fit. In contrast, regions with missing or extra particles are poorly fit. In these poorly-fit regions the residuals appear either as isolated missing particles, as isolated additional particles, or as a combination of the two – *e.g.* two nearby particles falsely featured as one large particle. In addition, optimizing the particle positions and radii usually causes some of the particles parameters to move to suspicious values, such as nearly-zero radii for particles that should not be added, particles with excessively large radii that really should be two particles, or particles positions placed well outside the image. We take advantage of the conspicuous differences in the residuals and of suspicious parameter values to correct any missing or mis-featured particles automatically. Our module for fixing missing and mis-featured particles contains two functions, one which we use when there are many missing and mis-featured particles, and a second function which we use to correct mis-featured particles that are tough to identify.

The first function proceeds in two stages. First, it removes any suspiciously large or small particles, or particles that are placed well outside the image. This removes most of the incorrectly added particles (false positives). Second, it runs a centroid algorithm on the fit residuals, and takes the 50 most promising particles and checks to see if they should be added to the model. The function does this by adding a particle at a typical radius (using the median radius of the current particles as the default) at the featured positions, optimizing only the particle's position, and keeping the new particle if the fit error reduces by at least a certain threshold; a threshold of $3\times$ the estimated noise level and 20% of the change in the fit's error suffices. The particle's radius is not optimized as changing the radius to a very small or very large value can occasionally reduce the error when the particle should not be added. The module iterates over these first two steps (removing suspicious particles and adding particles based on vacancies in the residuals) until convergence; at this point almost all of the particles are correctly identified. However, occasionally there are a few mis-featured particles which are not fixed by this procedure. For instance, sometimes two particles are very close to each other and are featured as one large particle. If this particle is not large enough to be identified as suspicious in part 1, then further iterations will not fix this mis-featuring. Even if the large particle is removed, the two particles may be close enough to be identified as only 1 particle by a centroid algorithm, and iterating the procedure will not fix the mis-featuring.

The second function fixes these tough mis-featured particles. It identifies local regions of the image that are poorly fit, by convolving the square of the residuals with a $5 \times 5 \times 5$ pixel-cube boxcar. It then removes all the particles in this poorly-fit region and repeatedly centroid-fits only the local region, checking and adding the featured particles one-by-one as before until the region is well-fit. With a combination of these two functions, our method is usually able to correctly identify every particle in the image. After all the particles have been added, the model is then optimized to completion.

D. Timing and performance

As discussed previously, the calculation of a single image scales with the number of pixels N as $N \log N$ due to the use of FFTs in our calculations. While the components of a model (platonic image, illumination, etc) may be calculated independently over a particular pixel several times, the dominant source of computational complexity is the convolution with the point spread function via FFTs. The time scale for optimization therefore would naively be $PN \log N$ where P is the number of parameters, since for each parameter a separate image update must be computed for calculate the gradient. Typically P is huge, $P \sim 10^4$, representing a huge prefactor in performance. However, essentially all of these parameters are computed over independent (slightly overlapping) sections of the image. Therefore, the optimization in reality takes $MN \log N$ where M is the number of global model components (platonic image, illumination field, PSF) and is relatively small.

Starting from scratch (no priors on any parameter), the optimization takes roughly 16-24 hours on a quad core i7 CPU for a large (512x256x64) confocal image. Due to the $N \log N$ scaling, smaller images can be analyzed from scratch much quicker. In fact, we recommend and often do employ small sections of images to first optimize the point spread function and exposure parameters that we later transfer to the full image for faster convergence. By analyzing a smaller image and transferring its global parameters to a large image, we can speed up this initial convergence to approximately 10 hours of computer time.

On the other hand, featuring is often used on many frames of a movie. In this case, we are able to transfer global parameters between frames and optimize on a new set of particles in roughly 1-3 hr on an i7. In the future we anticipate being able to offload inner-loop calculations on the GPU using CUDA for an improvement of roughly 8x but have not begun the implementation.

E. Source code

A complete implementation of this method is provided in a Python package called `peri`, whose source can be found online [21] along with extensive documentation about the particulars of its implementation. Additionally, it is available at `PyPI.org`, the central repository for Python packages outside of the standard library.

F. Available models and extensions

We view the theory of PERI and its corresponding software implementation as an encompassing framework and toolkit, which provide a tangible, guided practice for maximal precision feature extraction. However, we also emphasize that due the importance of accounting for the physics of image formation, no one model will perform well in all experimental scenarios.

At the time of writing this manuscript, the PERI software package implements only a select few imaging models based on usage patterns within our lab as well as common usage scenarios in the community which were easy to accommodate. In the main text, we discuss our primary use case of 3D line-scanning microscope images of nearly mono-disperse silica particles in a dyed fluid. However, with simple modifications, we are able to include several different variations on this model in our released software package. We discuss those here. Note that due to the modular nature of our software package the elements we discuss here can be combined in any fashion so the available number of additional models is actually a product over all individual elements in different categories.

1. Alternative PSFs

Since the line-scan PSF is implemented as a point-scan PSF integrated in along one dimension, we include the point-scan PSF with no additional effort within our source distribution. However, due to availability and time constraints, we have not directly verified its accuracy against an experimental system. The accuracy of the line-scan PSF indicates that implementation is most likely correct. With small modifications, a spinning disc confocal can be accommodated within our system just as the line-scan was created from the point-scan. It is not implemented as of the time of submission of this manuscript.

In our source distribution we also include various physically incorrect PSFs such as a 2D, 3D, and z-varying 3D Gaussian point spread function that can be used for testing purposes or improved convergence time.

2. Particle types and fluorescence

Currently PERI implements spherical particles of any polydispersity down to ~ 200 nm, limited by our ability to draw spheres smaller than a pixel size while maintaining volume and location accuracy. However, PERI is able to accommodate any polydispersity and has been tested with particle mixtures ranging from $a_{\mathrm{max}}/a_{\mathrm{min}}$ from ~ 1.1 to ~ 10 though in principle this is only limited by the pixel size on the smallest scale and window size on the largest. To remove the size limitations at the smallest scale a new type of model must be employed with a substantial deviation from our current confocal image model, which we may address in a later publication.

Another common imaging setup is for the particles to be fluorescently dyed rather than fluid. We include an implementation of this scenario in our source distribution but have not checked its accuracy as rigorously as the imaging system presented in this manuscript. There are several concerns which should be examined thoroughly before using in production, the primary being heterogeneous distribution of the dye either between particles or within a single particle. The problem of different particles being dyed a different amount can be fixed by augmenting the description of a particle with a dye concentration, in addition to its position and radius. If instead there are dye heterogeneities within a particle, then one could fit them using a generative model, perhaps composed of spherical harmonics on a per-particle level. On the other hand, if there is a systematic radial bias among all the particles then disentangling these systematics from PSF effects or the illumination field could become challenging. We leave these studies for a later publication as well.

3. 2D confocal images

Our software implementation natively supports 2D confocal images in addition to the 3D images studied in this manuscript. It is important to note that just like the 3D version, the particles outside of the image as well as coverslip must be included in the analysis. While 3D images give a sense of whether these particles are above or below the image, the nearly symmetric point spread function makes it difficult to determine direction in the case of 2D images. This introduces local minima in the fit landscape. However, there is a competing effect – if we are able to account for the signal due to the missing particles, it doesn't matter if we got their positions correct. Therefore, in practice, we can often fit 2D images without issue especially when the sample is nearly two dimensional.

Additionally, the resolution in these 2D systems scales as $\sigma_{\mathrm{2D}} \sim \sigma_{\mathrm{3D}}\sqrt{N_{\mathrm{3D}}/N_{\mathrm{2D}}}$ where σ_{3D} is given by Eq. 1 in the main text and N is the number of pixels that the particles occupies in the image. Given in terms of the particle radius at the optical section in pixels R, $\sigma_{\mathrm{2D}} \sim \sigma_{\mathrm{3D}}\sqrt{R}$. Therefore, for a 1 μm diameter particle at SNR = 20, the positional accuracy would go from ~ 1 nm to $\sim 2-3$ nm depending on the particle's position with respect to the viewing plane.

4. External models

Along with these included models and extensions, there exist other models in the development process that are not included in the software package at the time of publishing. In particular, other researchers are extending the PERI framework to work with scanning SQUID microscopy as well as STEM images for both image registration and particle featuring. The authors are in the process of developing a 2D brightfield microscope model as well.

5. Creating a new model

Given the modular nature of the software package, it is particularly easy to implement, optimization, and examine the performance of new models. In our documentation, we provide an extensive tutorial on the best practices to use when creating a model for a new imaging system not present in the official distribution. We encourage others authors to contribute their models back into the official package so that other researchers may benefit from their work.

VI. BENCHMARKS OF FEATURING ALGORITHMS

In the main text, we claim that PERI locates particles to within 3 nm, and that this constitutes an improvement of a factor of 10-100x over current methods. Clearly this statement implies that current methods locate particles to within 30-300 nm. In contrast, most current methods claim sub-pixel accuracy of about 10-30 nm. Occasionally,

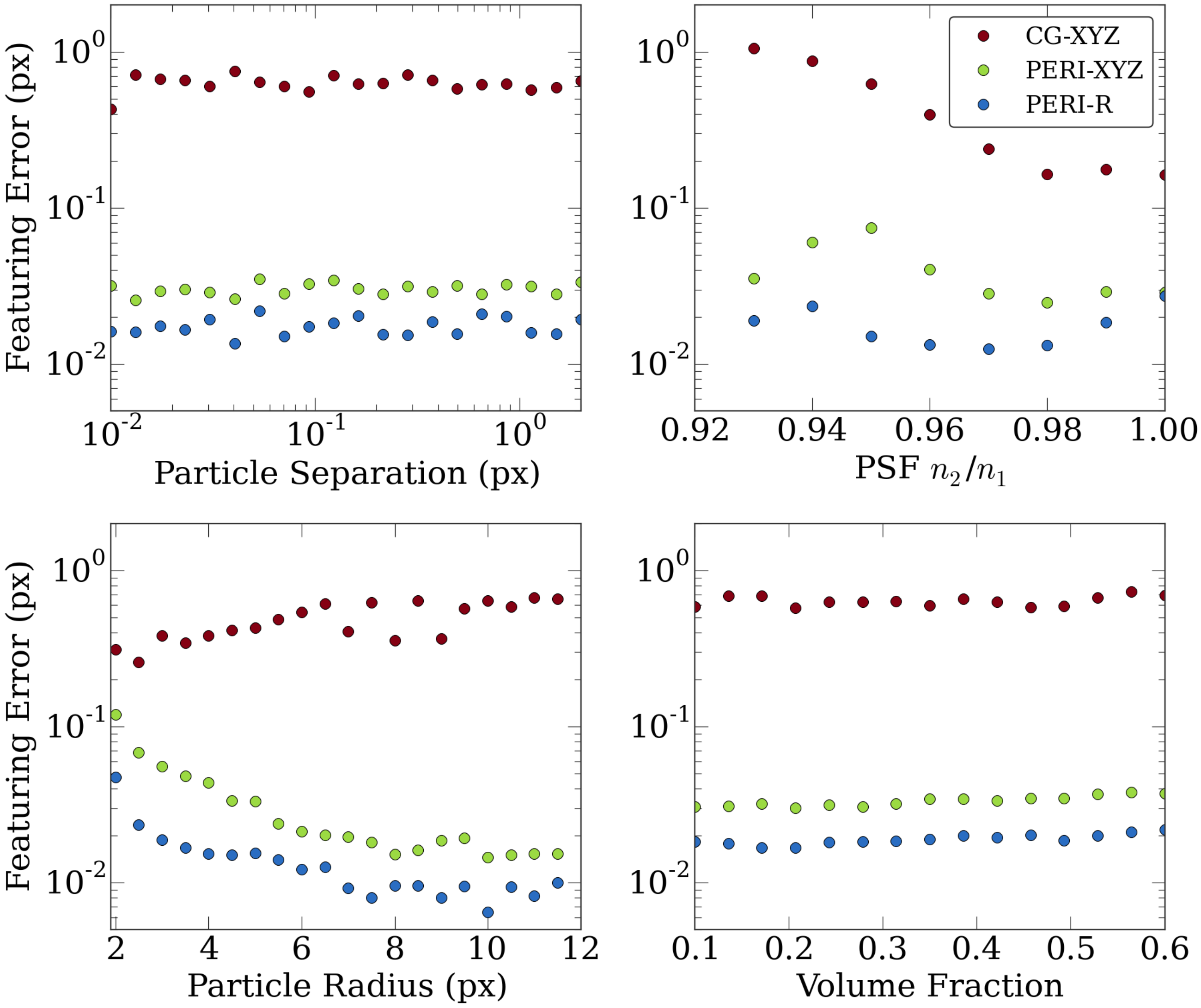

FIG. 15: **Accuracy benchmark**. We compare the featuring errors of PERI and a traditional centroid (Crocker Grier or CG) featuring method with the optimal featuring parameters. The panels show the featuring errors vs. particle separation (upper left panel), vs PSF aberration through the index mismatch n_2/n_1 (upper right panel), vs. particle radius (lower left panel), and vs. the suspension volume fraction (lower right panel).

model-independent checks, such as measuring particles' mean-square displacement and extrapolating to $\Delta t = 0$ [13], corroborate the standard methods' stated accuracies. However, these checks do not imply that the standard featuring methods provide 10-30 nm accuracies in all situations. By generating extensive data, we have found empirically that in dilute suspensions, with unaberrated point-spread functions, constant illumination fields, *etc*, that standard methods can locate particles accurately to within 30 nm or so – *i.e.* 10x worse, not 100x worse, than PERI. However, as the imaging conditions become more realistic (and more complicated), these other techniques do not perform well. In dense suspensions, we find that a straightforward application of a standard centroid algorithm frequently mis-identifies some particles with 200-300 nm (i.e. ≈ 2-3 px) errors, especially along the optical z-axis. These errors are almost completely systematic, arising from sources such as nearby particles whose diffraction pattern overlap, so the errors are not always easily measured by extrapolating the mean-square displacement or by fixing particles and measuring fluctuations in the fitted positions. In this section, we present some of our findings on the performance of standard algorithms in various imaging situations, and we compare their results to results from PERI.

A. Generated Data

We check PERI by benchmarking it against physically realistic image models, as shown in Fig. 15. For maximal realism, we generate these images with every model component in equation 2 as realistic as possible. We use our exact calculation for line-scanning confocal microscopes, with physical parameters expected from an experiment. From the structure of our fitted line-scan confocal images, we re-create a random illumination field that closely mimics the power spectrum of our actual confocal. We position the particles randomly, without placing them preferentially on the center or edge of a pixel. Since real images have particles that are also outside or partially inside the image, we generate the image on a large region before cropping to an internal region, resulting in edge particles and particles outside the field of view. [22]

We then fit these algorithms both with PERI and with traditional featuring algorithms. When we fit these images with PERI we start with initial guesses that are not near the correct parameter values, to ensure that our method is robust to realistic initial guesses. There are many heuristic algorithms and variants therein for locating colloidal particles [11, 13–18]. Ideally, we would compare our results to a standardized database with highly realistic images analyzed by each of the algorithms in existence. Without this database, it is difficult to accurately compare PERI to other methods, especially since many of the other methods have nuances and parameters that must be tuned by the user. Instead, we compare PERI against the most commonly used of these versions, a centroid-based method as implemented by Crocker and Grier [11] in the IDL language. All of these centroid algorithms require the user to select various parameters, such as a filter size for smoothing of the noisy image and a mask size for finding the centroid positions. As is well-known in the colloid community, using the incorrect parameters can result in significantly poorer results. To overcome any possible limitation from using the incorrect parameters, we *fit* all the possible parameters [23] in the Crocker-Grier (CG) algorithm and use only the ones that produces the best global featuring of the data, as compared to the correct particle positions. (Centroid methods do not accurately find particle radii). Needless to say, an actual experimenter does not have access to the ground truth or to the optimal parameters for the featuring. Moreover, even with these optimal parameters, the centroid algorithm frequently misses a large fraction of particles, even in simple images. As such, we view the centroid featuring errors as unrealistically optimistic and probably not attainable with centroid methods even by experts. The results of these comparisons are shown in Fig. 15.

When two particles are close, their images overlap due to the breadth of the point-spread function. This overlap causes centroid methods considerable difficulty. To compare the effects of PSF overlap on both PERI and CG featured positions, we generate an ensemble of realistic images with isolated pairs of particles at random orientations and at a fixed particle edge-to-edge separations. The upper-left panel shows these results for edge-to-edge separations from 0.01 px to 2.0 px, with a fixed noise scale of about 0.05 of the illumination amount. As the randomly-generated illumination fields vary from image to image, and the illumination varies from region to region within an image, there is not truly a global SNR for all of the images; the fluctuations in this SNR from image to image are the origins of the fluctuations in featuring error throughout Fig. 15. PERI features particles at the Cramer-Rao bound regardless of their separation. In contrast, even at large separations of 2 px, CG has significant errors due to particle overlaps.

Aberrations due to index mismatch significantly affect image quality and extracted particle locations. The upper right panel shows the effect of these aberrations on localizing isolated particles, as measured by the ratio between the index of refraction of the optics n_1 and of the sample n_2. Moving the ratio n_2/n_1 away from 1 increases aberration in the image. While increasing the aberrations in the lens negatively affects PERI's ability to feature particles, the localization accuracy always remains excellent. In contrast, CG methods perform poorly throughout, with extremely poor performance as the aberrations increase.

Since the CRB decreases with particle radius, we expect that increasing the particle radius should result in an increase in localization accuracy. The lower-left panel of Fig. 15 shows that PERI's precision improves with increasing particle radius. In contrast, the Crocker-Grier precision *worsens* with increasing particle radius. We hypothesize this arises due to the flat intensity profile near the center of a large particle, whereas a centroid method assumes that the intensity is peaked at the particle center. As a result, slight noise can significantly worsen a large particle's localization with centroid methods. Conversely, centroid algorithms improve for small particles, performing only 3× worse than PERI's localization accuracy for particles with radius 2 px. For particles small to the PSF size, the image is essentially a single peak, which centroid methods work well for.

Realistic images taken with confocal microscopes consist of particles randomly distributed, occasionally close together and occasionally far apart. To examine the localization in these images, we use a Brownian dynamics simulation to create a random distribution of particles at volume fractions from $\phi = 0.1$ to $\phi = 0.6$. PERI localizes particle positions and radii excellently in all of these images, as visible in the lower-right panel. In contrast, centroid methods perform uniformly poorly, with localization accuracies of approximately half a pixel. Interestingly, these centroid algorithms do not localize significantly worse for dense suspensions despite the presence of more close particles, although they do frequently fail to identify particles.

Finally, we check how the complexity of our synthetic data affects the accuracy of standard featuring methods. In

Polydispersity	Illumination field	Point spread function	Position error
0.0	Legendre 2+1D (0,0,0)	Identity	1.458
0.0	Legendre 2+1D (2,2,2)	Gaussian(x, y)	1.218
0.01	Barnes (10, 5), $N_z = 1$	Gaussian(x, y, z, z')	1.015
0.05	Barnes (30, 10), $N_z = 2$	Cheby linescan (3,6)	0.819
0.10	Barnes (30, 10, 5), $N_z = 3$	Cheby linescan (6,8)	1.085

TABLE II: **Crocker-Grier featuring errors**. We show the effect of image complexity on position error for the CG featuring method using synthetic data. Surprisingly, there is a non-monotonic behavior of error with complexity, hitting a maximum for highly striped images that don't vary strongly with depth.

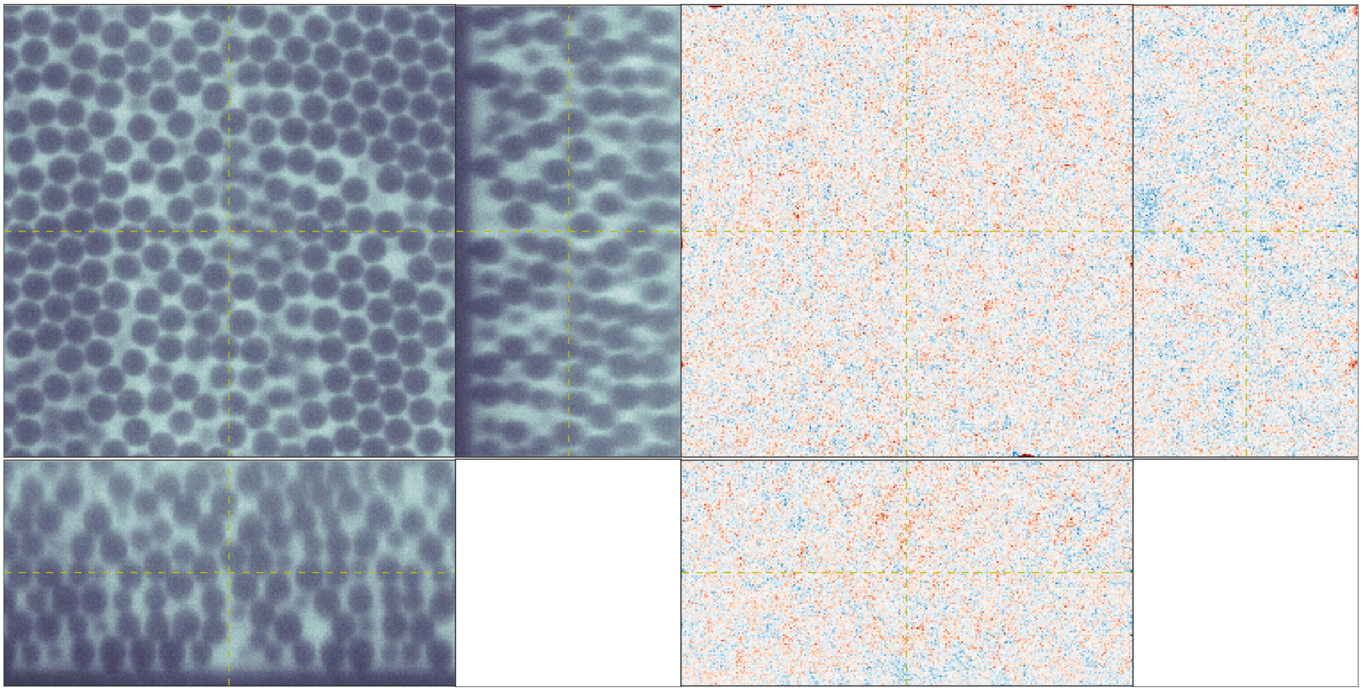

FIG. 16: **Featuring Stuck Particles.** The raw image of the 2 μm sample of fixed particles (left), and the residuals to the fit (right), shown in xy, yz, and xz cross-sections. Not only is the sample extremely dense, but as the image is quite deep the index mismatch between the sample and the confocal optics creates strong aberrations deep into the sample. Despite these complications, PERI is able to fit this complex image and to accurately locate particles in it.

Table II we see, surprisingly, that there is a non-monotonic relationship between positional error and image complexity, becoming optimal when there is significant striping in the image but little variation in depth. However, the rate of missing particles decreases significantly with simpler models and rising to as much as 40% for our most complex model images. The effective resolution of CG is never much smaller than a single pixel in these synthetic tests, most likely due to pixel edge biases.

B. Fixed Particles

Next, we check PERI on a sample of fixed particles. The sample is prepared by first making a dyed solution of 2 μm silica particles in an index-matching mixture of glycerol and water and loading the sample into a sample cell. At the edge of the sample we then add an equal amount of water-glycerol mixture saturated with salt (NaCl) and allow it to diffuse into the bulk of the sample over the course of a two weeks. As the salt diffuses in, it locally reduces the screening length and causes particles to strongly bind together. By letting the salt diffuse into the sample rather than mixing it in directly, the particles are able to sediment first before becoming fixed to each other, creating the dense sediment shown in Fig. 16. We then image these particles with a five-second delay between images and analyze the resulting images using PERI. The particle positions fluctuate by 2.9 nm, 1.7 nm, and 1.2 nm (median value) for z, y, and x, respectively, bounding the errors from above. (It is possible that some of the particles are not fixed to less than 2 nm.) We find radii fluctuations of 0.8 nm.

C. Other Model-independent Checks

In addition to testing the fit residuals for structure, analyzing the MSD of positions, and looking at the fluctuations of radii in time, we also looked at nearest neighbor isotropy. To do this, we constructed the 3D pair correlation function $g(\mathbf{r})$ using the extracted positions, scaled by the fit z_{scale}. In Fig. 17, we see in the left panel that the nearest neighbor displays no orientation dependence in the plane of the coverslip. The center panel of the same figure shows a cross section perpendicular to the coverslip, which shows the nearly 2D nature of the sample with approximately two layers of particles visible in the pair correlation. Finally, the right panel shows the distribution of in-plane azimuthal angles ϕ between nearest neighbors. We find no evidence of orientational biases due to pixelization present in the data [19]; the distribution of bond angles is isotropic as measured by a Kolmogorov-Smirnov test (p-value of 0.66).

As another model-independent check of PERI's ability to extract particle positions and radii at nanometer precision, we compared a sample of similar silica particles bonded to a substrate imaged both with SEM and our line-scanning confocal. While this comparison was able to place an upper bound on particle size measurements of 10-30 nm, we were greatly limited in by two confounding factors:

- *Charging effects* – Haloing from charge accumulation made precise measurements difficult with the SEM images. Using the furthest halo edge created particle overlaps while using our best estimate of the particle surface did not allow for any contacting particles. Using constrained fitting (enforcing the constraint that particles cannot overlap) yielded inconsistent results where the identified edges of particles varied in brightness from particle to particle. This variation meant that different parts of the charge-induced halo had been identified as the particle edge, leaving no consensus on the location of the true edge among the particle population.

- *Substrate deterioration* – The particle binding process left gouges in the coverslip that negatively impacted the particle positions and radii extracted from the confocal images using PERI. These gouges resulted in severe particle overlaps and poorly fit regions plainly visible in the fit residuals, which caused us to abandon the comparison. While in principle we could account for these divots using a more complicated generative model, their non-uniform appearance in the SEM images meant that they would be difficult to capture effectively in practice.

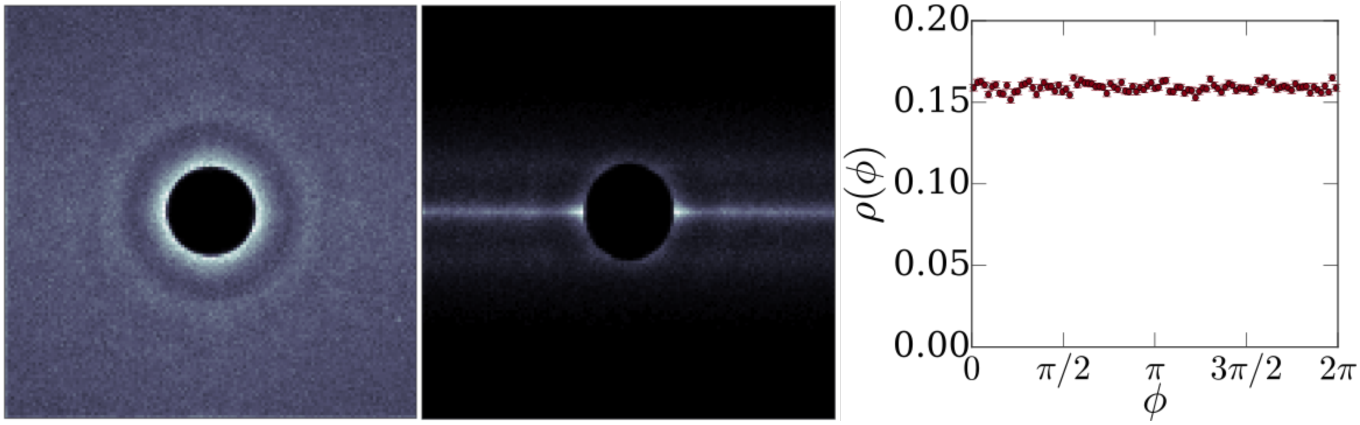

FIG. 17: **Nearest neighbor isotropy.** Left, center: Two different slices of the 3D $g(\mathbf{r})$ calculated from the experimental results presented in the main text. The (left) in-plane (parallel to the coverslip) cross-section shows no anisotropy in nearest neighbor bonds while (center) the out-of-plane cross-section highlights the nearly two dimensional nature of the colloidal sample. Right: The distribution of azimuthal bond angles ϕ between nearest-neighbors, with $\phi = 0$ corresponding to the x-axis. The bond angles show no sign of pixelization bias.

Finally, after verifying the particle parameters, we compared the global parameters to known microscope parameters. We measure the index of refraction of our suspension to be $n_2 = 1.437$ with a refractometer before starting the experiment. Using the known refractive index of the immersion oil and lens train, we expect $n_2/n_1 \approx 0.95$, with little uncertainty. Instead, we fit the index mismatch to be about 0.97, with tight error bounds on the fit. While reasonably close, these two values are excluded under the fit uncertainties. Why? One possible reason is a degeneracy in the description of the PSF. From Eq. 9 in the SI, it is easy to show, in the limit of weak aberrations (n_2/n_1 near 1), that the aberrations depend only on the product of $(n_2/n_1 - 1)$ and the distance from the optical interface, creating a soft direction. PERI finds the optimal fit for the PSF along this soft direction, fitting the effective position of the optical interface lower than it actually is. This trade-off presumably helps account for higher-order aberrations in the lens. (The best fit found by PERI fits the data better than the "sticker values" of the lens parameters.) Similarly, the aberrations in the lens also depend on the lens' acceptance angle or its NA, which causes the NA to drift slightly from its sticker value.

In general, the effects of model incompleteness on parameter values can be considered a form of "parameter renormalization" [20]. In this case, higher order aberrations are incorporated into the fit as part of a renormalized index of refraction, interface location, and other model parameters. While it is decidedly incorrect for precisely measuring properties of the microscope, the main issue for scientific application is more pointed – how do these renormalized parameters affect particle parameters? Generally speaking, global parameters tend to more strongly couple with other global parameters; however, there can be additional couplings to the particles through *e.g.* the particle radii. Couplings of this nature highlight the importance of independent parameter checks – SNR level of the fit residuals, number of overlapping particles, MSD extrapolation, and measuring fluctuations and drift in an individual particle's extracted radius. Though the fit value of the index of refraction is off by 2%, we know through our other checks that particle radii are off by no more than roughly 0.3%, the level of error that we quote throughout the paper.

[1] S. Hell, G. Reiner, C. Cremer, and E. H. K. Stelzer, "Aberrations in confocal fluorescence microscopy induced by mismatches in refractive index," J. Microsc. **169**, 391 (1993).

[2] S. L. Barnes, "A technique for maximizing details in numerical weather map analysis," Journal of Applied Meteorology **3**, 396 (1964).

[3] T. D. Visser and S. H. Wiersma, "Electromagnetic description of image formation in confocal fluorescence microscopy," J. Opt. Soc. Am. A **11**, 599 (1994).

[4] B. Zhang, J. Zerubia, and J.-C. Olivo-Marin, "Gaussian approximations of fluorescence microscope point-spread functions," Appl. Optics **46**, 1819 (2007).

[5] M. J. Nasse and J. C. Woehl, "Realistic modeling of the illumination point spread function in confocal scanning optical microscopy," J. Opt. Soc. Am. A **27**, 295 (2010).

[6] S. Hell, G. Reiner, C. Cremer, and E. H. Stelzer, "Aberrations in confocal fluorescence microscopy induced by mismatches in refractive index," Journal of microscopy **169**, 391 (1993).

[7] R. Wolleschensky, B. Zimmermann, R. Ankerhold, and M. Kempe, "High-speed scanning confocal microscope for the life sciences," in *European Conference on Biomedical Optics 2005* (International Society for Optics and Photonics, 2005) pp. 58600N–58600N.

[8] E. Dusch, T. Dorval, N. Vincent, M. Wachsmuth, and A. Genovesio, "Three-dimensional point spread function model for line-scanning confocal microscope with high-aperture objective," Journal of Microscopy **228**, 132 (2007).

[9] D. J. MacKay, *Information theory, inference and learning algorithms* (Cambridge university press, 2003).

[10] D. B. Allan, T. A. Caswell, and N. C. Keim, `\enquote{\bibinfo{title}{Trackpyv0.2},}\` (2014).

[11] J. C. Crocker and D. G. Grier, "Methods of digital video microscopy for colloidal studies," Journal of colloid and interface science **179**, 298 (1996).

[12] M. K. Transtrum and J. P. Sethna, "Improvements to the levenberg-marquardt algorithm for nonlinear least-squares minimization," arXiv preprint arXiv:1201.5885 (2012).

[13] Y. Gao and M. L. Kilfoil, "Accurate detection and complete tracking of large populations of features in three dimensions," Optics express **17**, 4685 (2009).

[14] R. Pathasarathy, "Rapid, accurate particle tracking by calculation of radial symmetry centers," Nat. Methods **9**, 724 (2012).

[15] R. Kurita, D. B. Ruffner, and E. R. Weeks, "Measuring the size of individual particles from three-dimensional imaging experiments," Nature Communications **3** (2012).

[16] M. Leocmach and H. Tanaka, "A novel particle tracking method with individual particle size measurement and its application to ordering in glassy hard sphere colloids," Soft Matter **9**, 1447 (2013).

[17] S. Andersson, "Localization of a fluorescent source without numerical fitting," Optics express **16**, 18714 (2008).

[18] S. M. Anthony and S. Granick, "Image analysis with rapid and accurate two-dimensional gaussian fitting," Langmuir **25**, 8152 (2009).

[19] P. J. Lu, M. Shutman, E. Sloutskin, and A. V. Butenko, "Locating particles accurately in microscope images requires image-processing kernels to be rotationally symmetric," Optics express **21**, 30755 (2013).

[20] M. K. Transtrum and P. Qiu, "Model reduction by manifold boundaries," Physical review letters **113**, 098701 (2014).

[21] Source code and tutorial at `http://www.lassp.cornell.edu/sethna/peri/index.html`

[22] Unless otherwise specified, we generate images with an index mismatch $n_2/n_1 = 0.95$, a ratio of fluorescent light to

excitation light energies of 0.889, an excitation wavelength of 488 nm, and a lens aperture acceptance angle of 1.173 corresponding to a 1.4 NA lens. The particles are 1 μm in diameter, with a pixel size used of 100 nm, and extend from a region from just above to ≈ 5 μm above the coverslip.

[23] We fit the x, y, z bandpass sizes for both the lowpass and hipass filters, the centroid size or diameter, the particle mass size “masscut”, the minimum particle separation, and a threshold below which pixels are ignored.