Chapter 2

Tracking of colloidal clusters with sub-pixel accuracy and precision

This chapter is published as C. M. van der Wel and D. J. Kraft, 'Automated tracking of colloidal clusters with sub-pixel accuracy and precision', J. Phys. Condens. Mat. **29**, 044001 (2017) DOI: 10.1088/1361-648X/29/4/044001

Abstract

Quantitative tracking of features from video images is a basic technique employed in many areas of science. Here, we present a method for the tracking of features that partially overlap, in order to be able to track so-called colloidal molecules. Our approach implements two improvements into existing particle tracking algorithms. Firstly, we use the history of previously identified feature locations to successfully find their positions in consecutive frames. Secondly, we present a framework for non-linear least-squares fitting to summed radial model functions and analyse the accuracy (bias) and precision (random error) of the method on artificial data. We find that our tracking algorithm correctly identifies overlapping features with an accuracy below 0.2 % of the feature radius and a precision of 0.1 to 0.01 pixels for a typical image of a colloidal cluster. Finally, we use our method to extract the three-dimensional diffusion tensor from the Brownian motion of colloidal dimers.

2.1 Introduction

Extracting quantitative information about the position and motion of features in video images is often key to understanding fundamental problems in science. For example, the tracking of colloidal hard spheres in three-dimensional confocal images has provided important insights into phenomena such as melting, crystallization, and the glass transition.[52–56] Biophysical experiments such as the investigation of cell mechanics by microrheology[57,58] or the measurement of single biomolecule mechanics using optical or magnetic tweezers[59] rely on the precise positional measurement of single colloidal particles. Moreover, the tracking of single proteins in live cells provided a powerful tool for understanding biological processes,[60,61] and eventually lead to the development of super-resolution microscopy techniques such as PALM and STORM.[62–64] Crucial for these studies is a method to extract trajectories of features from video images, which has been described extensively in colloidal science[65,66] as well as in single molecule tracking.[67–70]

Most single particle tracking algorithms have been designed for spherical features, as it is the most common type of signal. Recent developments in colloidal synthesis[43,71,72] provide means to create anisotropic particles, for example by assembling spheres in so-called colloidal molecules. Single particle tracking of these clusters of spheres is expected to provide insights into the role of anisotropy in for instance crystallization and diffusion.[40,73,74] As the basic building blocks of these studies contain closely spaced or partially interpenetrating spherical particles, a robust automated method is required to perform accurate particle tracking on partially overlapping features.

Automated methods for single-particle tracking follow roughly the following pattern: an image with features of interest is first preprocessed to remove background and noise, then single features are identified in a process called "segmentation", these feature coordinates are refined to sub-pixel accuracy, and finally the features are linked to the features in the previous image. Iteration of this algorithm over a sequence of images results in particle trajectories that can be used for further analysis. Although this method has proven itself as a robust and accurate method,[75,76] issues arise when features become so closely spaced that their signals overlap. Commonly, these issues are avoided experimentally by studying dilute systems, repelling particles, or model systems with very specific characteristics such as index-matched and core-shell fluorescent particles.[77,78] However, this is not always a possibility: therefore, issues from feature overlap persist in many measurements, especially involving colloidal clusters.

In particular, overlapping feature signals give rise to two complications: firstly, the segmentation step regularly recognizes two closely spaced features as one feature due to the overlap of signals. In order to identify the trajectories of closely spaced features completely, tedious frame-by-frame manual corrections are necessary, prohibiting the analysis of large data sets. In super-resolution microscopy methods, reported approaches to solve this issue are repeated subtraction of point-spread functions of detected features,[79] or advanced statistical models classifying merge and split events.[80] Notably, these tracking methods do not use all the available information: as the feature locations are known

in the previous frame, the segmentation of the image may be enhanced using the projected feature locations. This has been shown to improve segmentation significantly, for example in the tracking of biological cells.[81,82] Here we will present a fast and simple method for image segmentation that makes use of this history of the feature locations. We will test this method on artificial images and experimental data of colloidal dimers.

A second issue that arises when two feature signals overlap is that their refined coordinates will underestimate the separation distance. Especially the commonly employed centre-of-mass centroiding suffers from this systematic "overlap bias", leading to an apparent attraction between colloidal particles.[76,83] For fluorescence images, this issue can be addressed by least-squares fitting to a sum of Gaussians, which has been reported as a way to measure the distance between overlapping diffraction limited features.[84,85] Here, we will apply this method to images with features that are not diffraction limited. We will conduct systematic tests on the accuracy (bias) and precision (random error) of the obtained feature positions.

To demonstrate the automated segmentation and refinement methods, we will apply it to three-dimensional confocal images of a diffusing colloidal cluster consisting of two spheres and use the obtained trajectories to extract its diffusion tensor.

2.2 Methods

2.2.1 Segmentation

As our algorithm for single particle tracking is based on the widely employed algorithm by Crocker and Grier,[65] we will first introduce their algorithm and call it "CG algorithm". Throughout this work a Python implementation of this algorithm, Trackpy,[86] was used for comparison. The CG algorithm consists of four subsequent steps: preprocessing, feature segmentation, refinement, and linking. See Figure 2.1a for a schematic overview.

The preprocessing consists of noise reduction by convolution with a 1 pixel (px) sized Gaussian kernel and background signal reduction by subtracting a rolling average from the image with kernel size $2R + 1$. The length scale R is chosen just larger than the feature radius. The subsequent segmentation step finds pixels that are above a given relative intensity threshold and are local maxima within a certain radius S. The length scale S is the minimum allowed separation between particles. After the refinement step (see next section) the linking connects the features in frame i with features in frame $i - 1$ by minimizing the total displacement between the frames. Between two frames, particles are allowed to move up to a maximum distance L.

In this process, each frame is treated individually: only during the final step (linking), features are connected into trajectories. We rearranged this process so that the information about the particle locations in the previous frame is used already in the segmentation. This allows us to project the expected feature locations in consecutive frames and therefore increases the success rate of segmentation. See Figure 2.1b for a schematic overview. We describe our segmentation algorithm here using a minimal example

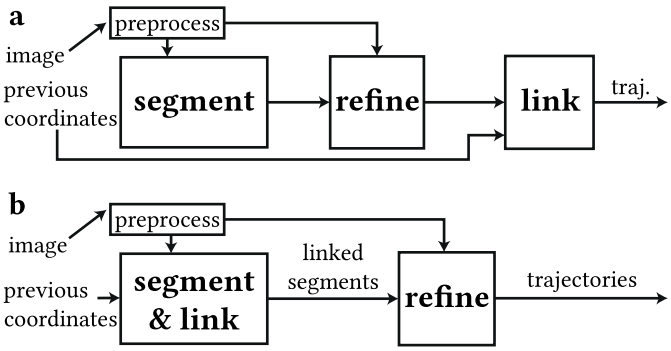

Figure 2.1. Schematic of the particle tracking of a single frame in (a) the CG algorithm and (b) the new algorithm. In the CG algorithm (a) the image is preprocessed and segmented. Starting from the segments and the preprocessed image, a refinement step is performed. Finally, consecutive coordinates are linked together with the coordinates in the previous frame. In our algorithm (b) the image is preprocessed and segmented, making use of the knowledge of the previous coordinates. The linked segments are refined afterwards to yield sub-pixel precision.

of two closely spaced features in two consecutive frames, which can be generalized to an arbitrary number of features in any number of frames. The technique is also valid in three dimensions, as demonstrated in section 2.3.3. See Figures 2.2a–c.

We will assume that feature finding and refinement was performed successfully on frame 1 (Figure 2.2d). Frame 2 is first subjected to grey dilation and thresholding step, just as in the CG algorithm. Because features are closely spaced in that frame, this leads to segmentation into only one single feature (Figure 2.2e).

Then a part of the linking step is executed: features are divided into so-called subnetworks. This is a necessary step in the CG algorithm to break the $\mathcal{O}(N!)$ sized combinatorial problem of linking two sets of N features into smaller parts. First, linking candidates are identified using a kd-tree.[86,87] Linking candidates for features in frame 1 are features that are displaced up to a distance L in frame 2 and vice versa. Then subnetworks are created such that all features that share linking candidates are in the same subnetwork. For a sufficiently large distance L, all features in Figure 2.2f belong to the same subnet: the feature in frame 2 is a linking candidate for both features in frame 1.

From the subnetworks, the number and estimated location of missing features is obtained "for free": if a subnetwork contains fewer particles in frame 2 than in frame 1, there must be missing features in its vicinity. To account for the possibility that a missing feature could connect two subnetworks, we combine subnetworks if they are less than a distance $2L$ apart in frame 1 whenever missing features are being located.

In order to estimate the location of the missing features, a region up to distance $L + S$ around the features in frame 1 is masked in frame 2 (dashed yellow line in Figures 2.2g–h). Subsequently, all already found features are masked up to a radius of S

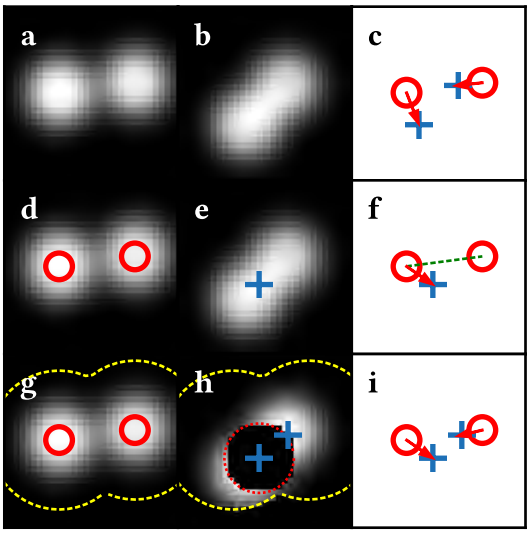

Figure 2.2. Artificial example to illustrate the integrated segmentation and linking step. In (a) and (b) two consecutive computer-generated frames are shown and in (c) the corresponding true feature locations, with the frame 1 features in red circles and the frame 2 features in blue crosses. Links are indicated by red arrows. In (d) frame 1 is shown again, overlaid with its feature coordinates in red circles and in (e) the result of the initial feature finding is indicated by a blue cross on top of frame 2. (f) The subnet is formed by the linking candidates. Additionally, the green dashed line denotes a distance between features that is less than $2L$. Therefore these features could belong to a single subnet via a missing feature. (g) Subsequently, a region of interest (dashed yellow line) is defined, comprising the pixels that are closer than $L + S$ to any feature in frame 1. This region is used to identify to which positions features in frame 1 could have moved in frame 2 (h). Also all features that were found already in frame 2 are masked out (dotted red line). These two operations enable the detection of the missing feature, which is then added to the subnet so that the linking can be completed (i).

(Figure 2.2h). This enables us to find local maxima that are further than distance S from all other features in frame 2 and closer than distance L from the features in frame 1. From the masked sub-image, local maxima are obtained again through grey dilation and thresholding. After this, feature selection filters can be inserted in order to select appropriate features, for example with a minimum amount of integrated intensity. Then the new feature is added to the subnetwork and linking is completed by minimizing the total feature displacement (Figure 2.2i).

By performing the linking during the segmentation process, additional information is taken into account: not only the present image is used to identify the features, but also the coordinates from the previous frame. Therefore, we expect a higher number of cor-

rectly identified feature positions for the combined linking and segmentation method. Because all the computationally intensive tasks were already present in the original algorithm, the execution time of our new algorithm was observed to be similar.

2.2.2 Refinement

Sub-pixel accuracy and precision are key features of single particle tracking. Although the size of a single pixel is diffraction limited to approximately 200 nm, localization precisions down to 1 nm have been reported.[69,75] These sub-pixel feature locations are obtained by starting from an initial guess supplied by the segmentation step, which is then improved in the so-called "refinement" step. Here, we will describe a general-purpose framework for refinement of overlapping features using non-linear least squares fitting to summed radial model functions.

We will compare this method to the centre-of-mass centroiding that is present in the CG algorithm.[65] For radially symmetric features, the feature position is given by its centre-of-mass. Due to its simplicity and computational efficiency, this method is a preferred choice for many tracking applications. In the centre-of-mass refinement, the centre coordinate $\vec{c}$ of the feature is obtained from the image $I(\vec{x})$, such that:

$$\sum_{\mathrm{dist}(\vec{x},\vec{c})\leq R} I(\vec{x})(\vec{x}-\vec{c}) = 0. \tag{2.1}$$

In order to obtain the centre-of-mass, a region $\mathrm{dist}(\vec{x},\vec{c}) \leq R$ is selected using the initial coordinate guess from the segmentation. In general, this is not necessarily close to the actual feature position: see for example Fig. 2.2e. Therefore, commonly an iterative centre-of-mass algorithm is employed[86] to be able to shift the region on which the centre-of-mass is computed. Throughout this work we will employ this iterative algorithm to find the centre-of-mass position.

Non-linear least squares fitting to a model function is conceptually different, since it goes beyond assuming only feature symmetry and requires knowledge on the feature shape. If image noise is uncorrelated and normal distributed, this method gives the maximum likelihood estimate of the true centroid. Although this assumption is not strictly valid,[66,69] the precision of this method is generally higher than the centre-of-mass method when the image is subject to noise.[75] By simultaneously fitting a sum of multiple model functions, this method can be extended to tracking multiple overlapping features.[84,85] We employ this approach here and formulate the feature model function F in the following way:

$$F(\vec{x},\vec{c},A,\vec{\sigma},\vec{p}) = \begin{cases} A\cdot f(r(\vec{x},\vec{c},\vec{\sigma}),\vec{p}) & \mathrm{dist}(\vec{x},\vec{c}) \leq R \\ 0 & \text{otherwise} \end{cases}, \tag{2.2}$$

$$r^2(\vec{x},\vec{c},\vec{\sigma}) = \sum_{j=1}^{D}\left(\frac{x_j - c_j}{\sigma_j}\right)^2. \tag{2.3}$$

Here, $\vec{x}$ is the image coordinate, $\vec{c}$ the feature centre, A its intensity, $\vec{\sigma}$ its radius, and f a model function of a single feature, which is a function of r and a list of parameters $\vec{p}$. The reduced radial coordinate r is defined for any number of dimensions D and allows for anisotropic pixel sizes through the vector nature of $\vec{\sigma}$. The feature model function is defined only up to distance R from the feature centre. With this definition, it is possible to use any function for f and apply it to images with different signal intensities and physical pixel sizes through the separate parameters A and $\vec{\sigma}$. In our tests, we keep $\vec{\sigma}$ constant and allow $\vec{c}$ and A to be optimized.

The model image is constructed by the summation of the individual features, which are each only defined within a region with radius R. This additivity is a good assumption for fluorescence microscopy techniques.[76] We add a fixed background signal B, which we keep constant within each cluster of overlapping features, but we allow it to vary between clusters to account for spatially different background values. For an image or video consisting of N features, the following "objective function" is minimized:

$$\sum_{\vec{x}} \left(I(\vec{x}) - B - \sum_{i=1}^{N} F(\vec{x}, \vec{c}_i, A_i, \vec{\sigma}_i, \vec{p}_i) \right)^2 . \tag{2.4}$$

The feature model function F is given in Eq. 2.2. In order to avoid discontinuities in f around $r = 0$, we omit points $\vec{x}$ from the objective function that are closer than 1 pixel from any feature centre $\vec{c}$.

If all features are separated by more than $2R$, this minimization can be separated into N single feature problems. However, when features have overlapping regions, their objective functions cannot be separated and have to be minimized simultaneously. We separate the full image objective function (Eq. 2.4) into groups ("clusters") using the kd-tree algorithm.[87] Each of the resulting cluster objective function is minimized using the sequential linear least squares programming (SLSQP) algorithm[88] interfaced through the open-source Python package SciPy.[89] This SLSQP algorithm allows for additional constraints and bounds on the parameters. We use bounds to suppress diverging solutions and constraints to, for example, fix the distance between two features to a known value. The optimizer is supplied with an analytic Jacobian of Eq. 2.4 to increase performance.

The here described framework of feature refinement in principle allows refinement of any feature that can be described by a radial function. Although less computationally efficient than the conventional refinement by centre-of-mass, it can take into account feature overlap and additionally allows for constraints on parameters.

2.2.3 Testing methods

The above described methods for single particle tracking were tested quantitatively on both artificial and experimental data. In order to model fluorescence images of colloidal particles, we chose three different artificial feature functions. Firstly we use the Gaussian function to describe diffraction-limited features:

$$f_{gauss}(r) = \exp\left[-r^2\right]. \tag{2.5}$$

Secondly, we describe colloidal particles that are larger than the diffraction limit by a hat-shaped function consisting of a solid disc with radius d surrounded with a Gaussian-decaying boundary:

$$f_{disc}(r, d) = \begin{cases} \exp\left[-\left(\frac{r-d}{1-d}\right)^2\right] & r \geq d \\ 1 & \text{otherwise.} \end{cases} \tag{2.6}$$

Thirdly, as a model for particles with fluorescent markers on their surface only, we define a ring shaped function with ring thickness t:

$$f_{ring}(r, t) = \exp\left[-\left(\frac{r - t - 1}{t}\right)^2\right]. \tag{2.7}$$

See Figure 2.3 for model features of the hat- and ring-shaped functions. The artificial features are defined in such a way that their size σ equals the radius at which the feature signal has decayed by a factor $e^{-1} \approx 0.37$ from its maximum value. Note that σ is not the same as the previously defined refinement radius R. As R should be chosen larger than the feature radius, we took $R = 2\sigma$ in our tests.

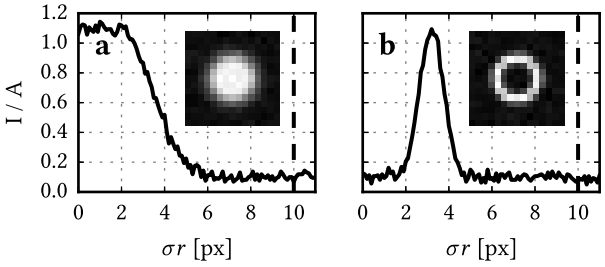

Figure 2.3. Radial intensity functions of (a) a hat-shaped and (b) a ring-shaped model feature in arbitrary units, generated with Equations 2.6 and 2.7 with parameters $\sigma = 4\,\text{px}$, $d = 0.5$, and $t = 0.2$. The insets show the corresponding single-feature images. Poisson distributed noise was added to each feature.

The feature is generated on a pixel grid from a randomly generated sub-pixel location. Unless stated otherwise, we chose the feature size σ to be 4 px. Depending on the application, the physical size of a single pixel typically ranges from 100–500 nm. The feature maximum A is chosen at 160 and unless stated otherwise, $d = 0.5$ and $t = 0.2$. Images were discretized to integer values and a Poisson distributed, signal-independent

background noise with a mean intensity of $N = 16$ was added to each image. The corresponding signal-to-noise ratio S/N is defined as A/N. Each refinement test was performed on 100 images having two overlapping features with a given centre-to-centre distance and random orientations. In order to ensure that the choice of initial coordinates did not affect the refined coordinate, we generated the initial coordinates randomly within 0.5σ from the actual coordinate.

Experimental measurements on colloidal particles were performed with an inverted Nikon TiE microscope equipped with a Nikon A1R confocal scanhead. For the two-dimensional diffusion measurements, we used a 20× objective (NA = 0.75), resulting in a physical pixel size of 0.399 µm in x- and y-directions. The images were recorded with Galvano mirrors acquiring images of 64 × 64 pixels at 15.4 Hz. For the three-dimensional measurements, a 100× (NA = 1.45) oil immersion objective was used, resulting in an xy pixel size of 0.166 µm. A calibrated MCL NanoDrive stage enabled fast z stack acquisition with a z step size of 0.300 µm. As the objective immersion liquid ($N_D = 1.515$) is closely matched with the sample solvent ($N_D = 1.49$), this step size equals the physical pixel size in z direction within an error of 5 %.[90] We acquired 5.13 three-dimensional images per second with a size of 512 × 64 × 35 pixels in xyz, respectively.

For two-dimensional diffusion measurements we used samples consisting of partially clustered TPM (3-(trimethoxysilyl)propyl methacrylate) colloids with a diameter of 1.94 ± 0.09 µm containing an RITC (rhodamine B isothiocyanate) fluorescent marker, as described in Chapter 7. Because of their density difference with water, the particles were confined to the coverslip by gravity, enabling two-dimensional tracking.

The samples for three-dimensional measurements consisted of core-shell RITC labelled PMMA (polymethylmethacrylate) colloidal clusters that were synthesized via an emulsification-evaporation method according to ref. [71]. The average distance between the two constituent spheres of diameter 1.87 ± 0.06 µm in a cluster is 1.58 ± 0.12 µm, determined by scanning electron microscopy using an FEI NanoSEM at 15 kV. The clusters were both index and density matched using a mixture of cyclohexyl bromide and *cis*-decalin in a weight ratio of 72:28 and imaged in a rectangular capillary, similar to experiments described in ref. [91].

The Python code on which this work is based is available online in the package Trackpy,[86] that is available through Conda as well as through the Python Package Index. All tests described in this work are implemented as "unittests" that ensure the correct functioning of the code on each update.

2.3 Results and Discussion

2.3.1 Segmentation and Linking

As described in the Methods section, the integrated segmentation and linking step extends the frame-by-frame segmentation used in the CG algorithm in such a way that it makes use of the history of feature locations. In order to test the effect of our exten-

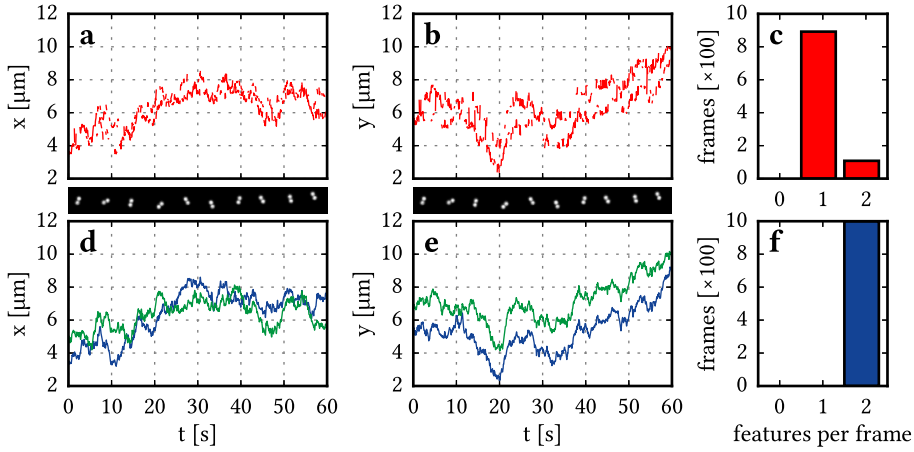

Figure 2.4. Segmentation of an experimental 2D video image of a colloidal dimer. In (a) and (b) the x and y coordinates obtained using the CG algorithm are shown. The corresponding histogram of features per frame is displayed in (c). As the two features were identified correctly in only 11 % of the frames, trajectories could not be identified. In (d) and (e) the trajectories were obtained using the integrated segmentation and linking algorithm. As all frames had two features (see the histogram in (f)), trajectories were identified completely. In these plots, coordinates were refined using least-squares fitting to a sum of Gaussians. A video of this comparison is available online.

sion, we compared the segmentation in the CG algorithm with our integrated segmentation and linking on experimental video images. As a distinct example of overlapping features, we analysed a single colloidal dimer, which consists of two permanently connected spheres. The dimer exhibits Brownian motion in two dimensions. The identified trajectories for 1000 frames are displayed in Figure 2.4.

The CG algorithm identified two features in only 11 % of the frames, resulting in short disconnected trajectories that appear to hop between two feature locations. Apart from that, the single features were often localized at a position in between the two features, which compromises the refinement step. The here described extension of segmentation improves the feature identification significantly: by taking into account the history of the feature positions, we detect two features in every frame.

As we have refined the feature positions using the described least-squares fitting method, we are also able to compute the average distance between the particles. We find $2.04 \pm 0.03\,\mu m$. As the dimer consists of aggregated spheres, we can compare this directly to the average particle diameter ($1.94 \pm 0.09\,\mu m$). Due to the size polydispersity of the sample, we cannot use this to quantify the bias in the feature positions. The precision can however be estimated using the spread in the measured separation lengths: we find 0.08 px. Part of this is stemming from a physical effect, as the dimer can slightly tilt in

the third dimension.

The here described extension of segmentation increases the number of correctly segmented features significantly. It has to be noted though that the segmentation of the first frame is not enhanced by our method because of the lack of information on the previous feature positions. Generally, there is a start-up period of a few frames in which the number of correctly segmented features increases. The length of this start-up period can be estimated from the probability of correctly identifying a dimer in absence of information about the previous positions: in our case, this probability is 11 %. This corresponds to a probability of 90 % of having at least one correctly segmented frame in the first 20 frames. These potentially incorrectly tracked frames can be ignored for most tracking applications. For cases where the first frames are relevant, the algorithm may be run backwards from the first correctly segmented frame.

2.3.2 Refinement

After the segmentation step the sub-pixel position is obtained in the refinement step. In this section we will analyse the effect of signal overlap on the accuracy and precision in the refined feature coordinates using both centre-of-mass and the here described least-squares fitting to sums of model functions. We define the accuracy or bias as the mean difference between the measured and the true value. The precision is the random deviation around the measured average, which we calculate with the sum of the root of squared deviations from the measured average.

Firstly, we took two Gaussian-shaped features (Eq. 2.5) with radius $\sigma = 4\,\mathrm{px}$ and varied their spacing between 1.5σ and 5.5σ. See Figure 2.5. The deviations of the obtained positions are measured parallel and perpendicular to the line connecting the two actual feature positions. We observed no bias in the perpendicular coordinate for either refinement method. For the parallel coordinate, however, we found a clear difference between the two approaches: in centre-of-mass centroiding, the parallel coordinate was negatively biased because of feature overlap, meaning that the distance between the two overlapping features was systematically underestimated. In least-squares refinement, this bias was not present and the features were located with sub-pixel accuracy for all particle spacings.

The negative bias for centre-of-mass centroiding has been described before[83,92] and is a consequence of the method: if two features overlap, each of the features obtains extra intensity on the inside of the dimer. This bias increases in magnitude with decreasing particle separation, until both features are detected precisely in between the two actual positions. The bias increases also with increasing mask radius R, as shown in Figure 2.5.

Apart from this negative bias, we observed a longer ranged positive bias. This effect has its origin in the preprocessing. For centre-of-mass centroiding, it is vital that any constant image background is subtracted. This is conventionally achieved by subtracting a rolling average of the image with box size of typically $D_{bg} = 2R + 1$.[65] Although this method has proven to be robust for background subtraction, it also introduces a skew in the feature signals when features are closer than $\ell + D_{bg}$ (see Figure 2.9 on page 29).

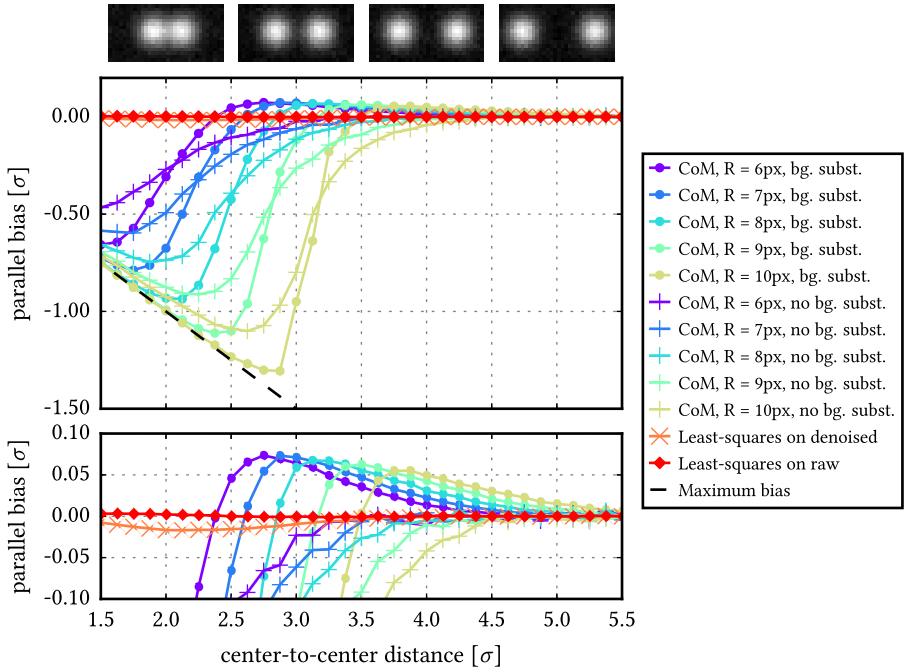

Figure 2.5. The effect of feature overlap on the bias in the parallel coordinate. The bias is negative when features appear too close together. In both graphs, the bias in the parallel coordinate as a function of the centre-to-centre distance is shown, for two Gaussian features with $\sigma = 4\,\text{px}$ and signal-to-noise ratio S/N = 10. The bias for the centre-of-mass (CoM) refinement is shown for mask radius R from 6 to 10, both with rolling average background subtraction (denoted with dots) and without (denoted with crosses). The bias for the least-squares fitting to a sum of Gaussians method is shown in orange tilted crosses for the case that the fitted image was denoised first. Direct least-squares fitting of the raw image is shown in red diamonds. The dashed black line denotes the bias at which features are detected precisely in between the two actual feature positions. The insets correspond to centre-to-centre distances of 2σ, 3σ, 4σ, and 5σ.

Here, ℓ is the typical feature diameter. From this we conclude that it is important not to use a rolling average background subtraction in order to accurately track features that are spaced closer than $\ell + D_{bg}$. If the background subtraction was omitted, the positive bias was indeed not observed, as can be seen in Figure 2.5. In order to account for the background signal in the least-squares fitting algorithm, we introduced a background variable B in the objective function (Eq. 2.4) instead.

The least-squares fitting to sums of model functions is clearly better able to provide an unbiased localization of the feature: the (absolute) bias stayed below 0.2 % of the

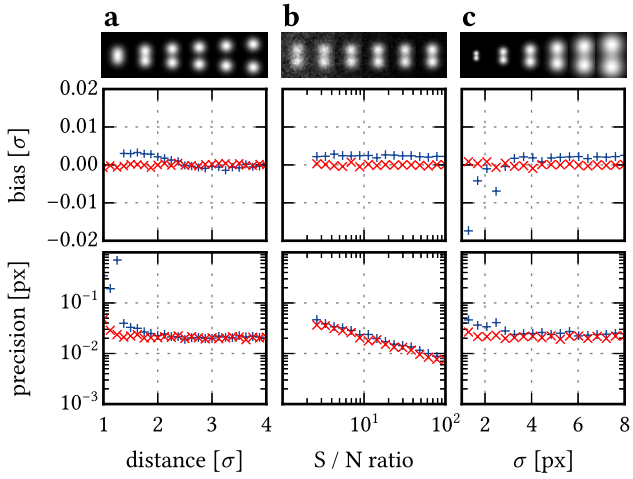

Figure 2.6. Localization errors of Gaussian-shaped features. The top row presents the computer generated model features, the middle and bottom row show the mean deviation (bias) and the root of the central variance of the deviations (precision), respectively. The data is separated into the error parallel (upright blue crosses) and perpendicular (tilted red crosses) to the line connecting the true feature positions. Unless stated otherwise, a feature size $\sigma = 4\,\text{px}$, a feature centre-to-centre distance of 2σ, and signal-to-noise ratio S/N = 10 were employed. (a) Overlapping features were localized with bias below 0.004σ and sub-pixel precision, for a feature separation distances larger than 1.4σ. (b) Above a signal-to-noise ratio (S/N ratio) of 2, bias was independent of noise and precision improved with increasing S/N ratio. (c) Below a feature size of 2 px, the (absolute) bias increased steeply. The precision was independent of the feature size.

feature radius for all separation distances. We observed that denoising the image before least-squares fitting causes a bias of up to 0.02σ, depending on the feature separation. This is because denoising adds a correlation between neighbouring pixels, while least-squares assumes an absence of correlation between datapoints. Therefore we conclude that direct fitting of a raw image gives the most accurate results.

Secondly, we analysed the bias and precision in the coordinates of overlapping Gaussian features, while systematically varying the particle spacing, signal-to-noise ratio, and size. See Figure 2.6. In all cases, we observed no bias in the perpendicular coordinate, as is expected from the symmetry of the dimer. Also, the precision for the perpendicular and parallel directions were in close agreement. As shown in Figure 2.11 on page 30, the bias is proportional to the feature size σ, while the precision is independent of σ. Therefore, we report the bias in units of σ, and the precision in units of pixels (px).

We observed accurate and precise refinement down to particle separations of 1.4σ: at shorter separation distances, the algorithm was not able to separate the features (see Figure 2.6a). In Figure 2.6b, it can be seen that the signal-to-noise (S/N) ratio does not

influence the bias, while the precision improves with increasing S/N ratio. At S/N < 2, the optimizer sometimes diverged and yielded random results. This failure of least-squares fitting was reported already for S/N < 4 by Cheezum and co-workers.[75] As the SLSQP minimization allows for bounds on the feature parameters, we were able to suppress the diverging solutions by limiting the displacements of centre coordinates to the mask size R. This enhancement enabled us to also use the least-squares method for $2 \leq$ S/N < 4.

In Figure 2.6c and Figure 2.11, it can be seen that the bias in the parallel coordinate is a constant fraction of the feature size (0.2 %). Although this bias is negligibly small, it is still larger than the bias in the perpendicular coordinate, showing the influence of particle overlap. Below a feature radius σ of 2 px, we observed that the bias increased steeply. Presumably there is not enough information in these small features to accurately determine the two positions.

As colloidal molecules are often larger than the diffraction limit, their feature shape is typically not Gaussian. We assessed the effect of the mismatch between a non-Gaussian feature and a Gaussian fit function by gradually changing the non-Gaussianness of the model feature, using hat- and ring-shaped model features as described by Eqs. 2.6 and 2.7. See Figures 2.7a and 2.10a (page 29). The observed precision in the refined position of the overlapping hat- and ring-shaped features was roughly independent of the mismatch between feature and fit function, but deteriorated at solid disc size $d > 0.8$ or ring thickness $t < 0.5$. The bias increased for increasing mismatch between feature and fit function up to a maximum of 0.06σ. Although a Gaussian fit of a non-Gaussian feature may appear to be very precise, we conclude here that it is systematically biased due to the mismatch between feature and fit function.

In order to obtain more accurate positions for non-Gaussian shaped features, we exploit the ability of our algorithm to accept any fit function. In Figure 2.7b–d we fitted the hat-shaped features with hat-shaped functions, and in Figure 2.10b–d ring-shaped features with ring-shaped functions. In both cases, we observed a bias below 0.2 % of the feature radius. The precision even increased for increasingly less Gaussian like features. We hypothesize that this is caused by the steeper gradients at the edges of hat- and ring-shaped features, which lead to a better definition of the position. The other trends in precision are similar to Gaussian-shaped features: the observed constraints are $S/N \geq 2$ and $\sigma \geq 1.8$ px for hat-shaped and $S/N \geq 2.5$ and $\sigma \geq 1.8$ px for ring shaped features.

To summarize, we observed that least-squares fitting to sums of model functions is able to accurately refine the location of overlapping features. The negative bias of multiple pixels present in centre-of-mass centroiding is reduced to less than 0.2 % of the feature radius if the feature radius is above 2 px and S/N ratio above 2. The least-squares algorithm is robust against large deviations of the initial location guess, which we tested for deviations up to 50 % of the feature radius. This makes fitting to sums of model functions an appropriate method for refining overlapping features, given that the size and noise constraints are met. Although the Gaussian fit function was able to fit hat-shaped and ring-shaped features with precisions compared to Gaussian-shaped

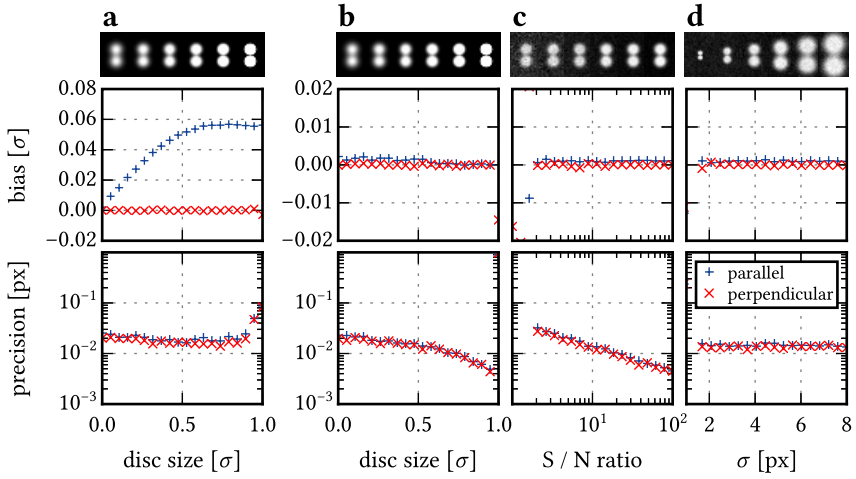

Figure 2.7. Localization errors of hat-shaped features (see Eq. 2.6), which are models for colloidal particles with a size larger than the diffraction limit. Tests were done using Gaussian-shaped (a) and hat-shaped (b)-(d) fit functions. (a) For an increasing solid disc size d, the bias increased up to 0.06σ due to an increasing mismatch between the feature and the (Gaussian) model function. The precision started to decrease above a relative disc size $d = 0.8\sigma$. (b) When a hat-shaped model function was used, the bias improved such that it stayed below 0.002σ for all d. It slightly decreased with increasing d. The precision increased with d. (c) The bias was independent of S/N, and the precision increased with increasing S/N ratio, for S/N values above 2. (d) The bias was proportional to σ, and the precision was independent of σ, for $\sigma \geq 1.8$ px.

features, we showed that the bias in fact increases to up to 6 % of the feature radius. By using more appropriate fit functions, this bias can be reduced to 0.2 %, as we showed for hat- and ring-shaped features.

As described by Jenkins *et al.* [76], it is possible to experimentally obtain an average feature shape and successfully use this for feature refinement of single features. If a continuous function is available that describes the average feature, this could be used directly in our framework for least-squares minimization, extending their technique to any number of overlapping features. A different approach of tracking overlapping particles should also be mentioned. Tracking only the non-overlapping part of colloidal dimers has been reported using a mirroring technique[93] or cross-correlation.[94] These techniques rely on a significant fraction of the feature that is not influenced by feature overlap, and are therefore mainly applicable to colloidal dimers.

We note here that from the objective function (Eq. 2.4), not only the minimum, but also the uncertainty in the minimum could in principle be obtained. This computation would provide a per-particle measure for the positional uncertainty, which is important especially for inhomogeneous images. Such a computation would require mapping out

the multidimensional contour at which the objective function $\Delta\chi^2 = \min(\chi^2)$,[76,95] which is beyond the scope of this chapter.

Although the reported accuracy of 0.002σ is sufficient for most applications, a further improvement could be reached by maximizing the log-likelihood corresponding to Eq. 2.4 instead of using the direct least-squares minimization. For single features, using a maximum likelihood estimator has been proven to give a more precise estimate of the true feature positions.[70,96]

2.3.3 Constrained least-squares

If additional information about the tracked features is available, constraints can be applied to increase tracking accuracy. In our framework for least-squares optimization of summed radial model functions, any combination of parameters in the image model function (Eq. 2.2) can be constrained by equations of the following form:

$$g(P_n) = 0 \qquad \text{or} \qquad g(P_n) \geq 0. \tag{2.8}$$

Here, g is a function and P_n is an array consisting of all parameters of features that are in a cluster of size n. We demonstrate the use of constraints here using colloidal dimers with known distance between the two constituent spheres. Using our algorithm we automatically tracked 1006 out of 1170 recorded frames. A constraint was chosen such that the distance between the constituent spheres equals the average distance measured on SEM images (1.58 µm). A video of the resulting tracked three-dimensional images is available online.

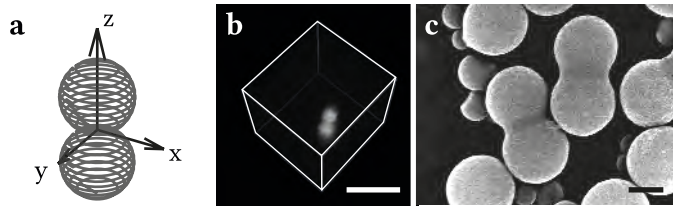

Figure 2.8. Images of colloidal PMMA dimers. (a) The coordinate system corresponding to the diffusion tensor originates from the point of highest symmetry. (b) A typical three-dimensional confocal image that is used for the particle tracking. (c) A representative Scanning Electron Micrograph of the employed colloidal dimers. The scalebars in (b) and (c) denote 5 µm and 1 µm, respectively.

As the shape of a colloidal cluster is anisotropic, the short-term diffusion of such a particle is also anisotropic: for example, a dimer experiences a lower hydrodynamic friction when moving along its z-axis, compared to when moving along its x-axis. In general, the dynamics of any Brownian object is described by a symmetric second-rank tensor of diffusion coefficients, consisting of 21 independent elements.[91] We chose the point of highest symmetry as the origin of the cluster based coordinate system and

Table 2.1. Tensor of dimer diffusion coefficients, averaged over time differences (lag times) of 0.2, 0.4, and 0.6 s. The coordinate system is defined in Figure 2.8. The translational coefficients are given in units of 10^{-3} μm^2 s^{-1}, the rotational coefficients in units of 10^{-3} s^{-1}, and the rotation-translation cross terms in units of 10^{-3} μm s^{-1}. Because rotation around the z-axis cannot be measured for a dimer, we omitted the corresponding elements. The error denotes the 95 % confidence interval estimated using a bootstrap algorithm.

	x	y	z	θ_x	θ_y
x	61.6±4.0	-0.9±2.8	-0.4±3.1	0.0±1.3	-0.4±1.3
y	-0.9±2.8	60.8±3.8	-0.7±3.0	-0.4±1.3	-0.2±1.4
z	-0.4±3.1	-0.7±3.0	65.2±4.2	-0.0±1.3	-0.4±1.4
θ_x	0.0±1.3	-0.4±1.3	-0.0±1.3	12.5±1.1	-0.2±0.7
θ_y	-0.4±1.3	-0.2±1.4	-0.4±1.4	-0.2±0.7	13.4±1.1

aligned the z-axis with the long axis of the dimer, so that all off-diagonal terms in the diffusion tensor are zero. See Figure 2.8a. In order to compute particle displacements for the diffusion tensor, any time difference can be used as long as it is short enough to preserve the orientational information of the cluster. We computed diffusion tensors for lag times of 0.2, 0.4, and 0.6 s. The resulting averaged diffusion tensor reflects the symmetry of the dimer and can be seen in Table 2.1.

In line with previous results from holographic microscopy measurements,[97] we observed that the translational diffusion constant along z is higher than the translational coefficient along x and y. These results illustrate that our new tracking algorithm is able to compute quantitative information from microscopy images of colloidal clusters without the need of manual corrections.

2.4 Conclusion

We have presented a new algorithm for single-particle tracking that enables automated tracking of overlapping features with high accuracy and precision. It is based on a the well-known algorithm developed by Crocker and Grier[65] and implements two improvements. First, by exploiting the information obtained from the linking already in the segmentation stage, we were able to use the history of the feature positions to obtain segmentation with significantly fewer mistakes. In a test on two-dimensional experimental data of dimers, all frames were segmented correctly, while the conventional algorithm correctly segmented only 11 % of the frames.

The second improvement consists of a method for sub-pixel accurate localization of overlapping features. The conventional centre-of-mass refinement is unable to find unbiased feature locations: signal overlap results in a negative bias if the feature separation distance is below the mask diameter, and the commonly used rolling average background subtraction imposes a positive bias already at separation distances below approximately 1.5 times the mask diameter. We reach sub-pixel accuracy and precision

by least-squares fitting the unprocessed images to sums of radial model functions.

Firstly, we tested Gaussian-shaped model features with varying separation distance, signal-to-noise ratio, and feature size and found an accuracy of less than 0.2 % of the feature radius, for separation distances above 1.4σ, S/N ratios above 2, and feature radii above 2 px. The precision depends on the S/N ratio only. Secondly, we showed that non-Gaussian features can be tracked with roughly similar precision as the Gaussian features, however the accuracy degrades to 6 % of the feature radius due to the mismatch between feature and fit functions. Thirdly, we showed that significantly better accuracies were obtained by using more appropriate feature models, such as hat- and ring-shaped functions.

We demonstrated the use of constraints in least squares fitting using experimental three-dimensional image sequences of colloidal dimers. Trajectories from 86 % of all frames were readily obtained without any manual refinement. From this, the diffusion tensor was reported and found to accurately reflect the particle symmetry.

With the described method, two problems are solved that are encountered when employing conventional tracking methods on overlapping features. Firstly, the need for case-to-case optimization or manual reparation of tracks is significantly reduced. Secondly, by employing least squares fitting to summed model functions we found that the bias of the centre-to-centre separation distance is 0.2 % of the feature radius in the worst case, which clearly outperforms the centre-of-mass centroiding. Our method provides accurate automated tracking of videos containing overlapping features with minimal need for manual adjustments.

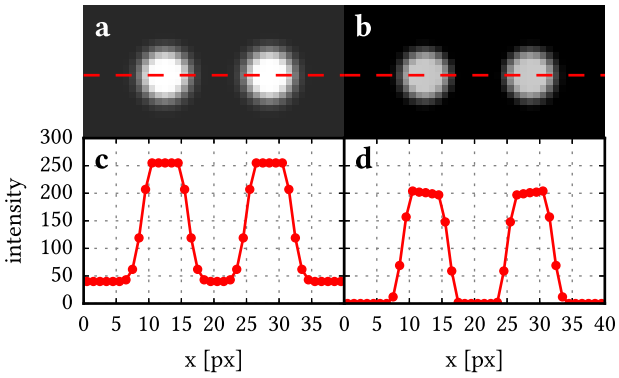

Figure 2.9. Illustration of the positive bias due to background subtraction. If the image background is non-zero (a), it can be subtracted using a rolling average resulting in a perfectly black background (b). However, in the image cross sections (c) and (d), it can be seen that the rolling average also results in a skew of the feature shapes, which gives an outwards directed bias when locating the feature positions.

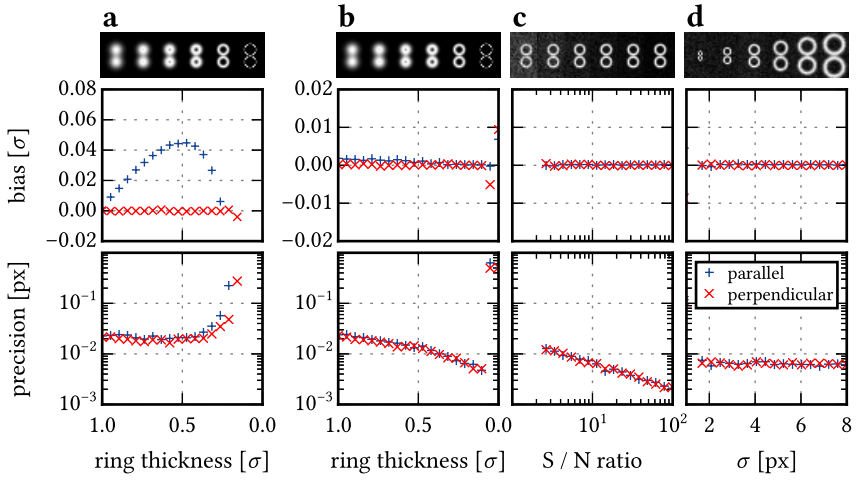

Figure 2.10. Localization errors of ring-shaped features (see Eq. 2.7). These features are models for colloidal particles with surface-bound fluorophores. Tests were done using Gaussian-shaped (a) and ring-shaped (b)-(d) fit functions. (a) Using a Gaussian fit function, the bias increased up to 0.05σ. Below a relative ring thickness t of 0.2σ, the refinement diverged. The precision started to decrease below $t = 0.5\sigma$. (b) When a ring-shaped model function was used, the bias stayed below 0.002σ for $t > 0.1\sigma$. The refinement diverged for too thin rings ($t \leq 0.1\sigma$). The precision increased with t. (c) The bias was independent of S/N, and the precision increased with increasing S/N ratio, for S/N values above 2. (d) The bias was proportional to σ, and the precision was independent of σ, for $\sigma \geq 1.8$ px.

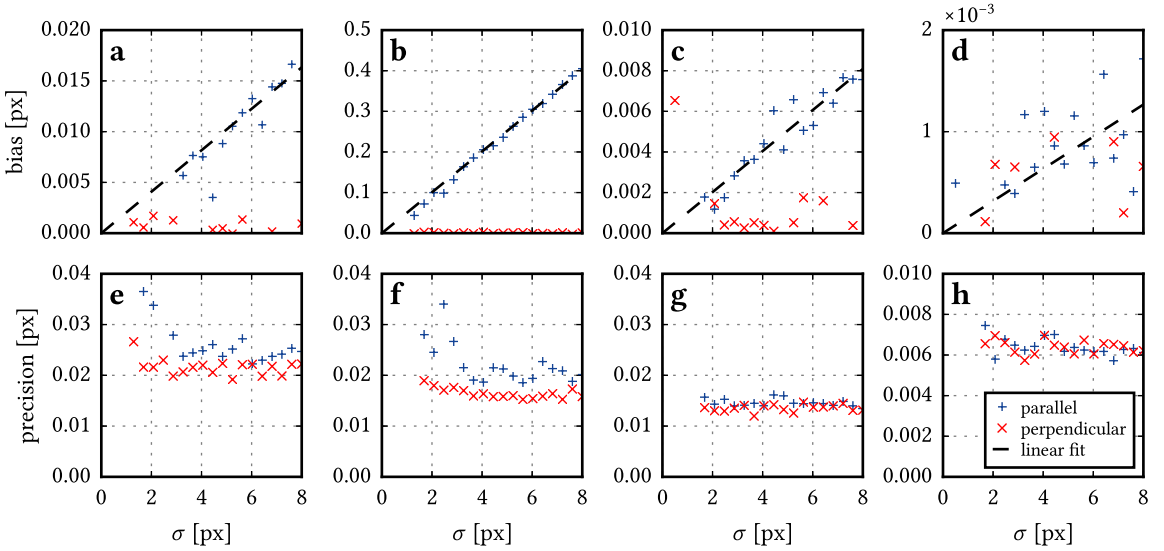

Figure 2.11. Trends of bias and precision with feature size σ. In all cases, the bias was proportional to σ, while the precision was independent of σ. We employed (a) a Gaussian feature to refine with a Gaussian fit function, and (b) a hat-shaped feature with a Gaussian fit function. In (c), a hat-shaped feature is refined with a hat-shaped fit function, and in (d) a ring-shaped feature with a ring-shaped function. In (e)-(h), the corresponding precisions are shown.