Nanoscale

PAPER

View Article Online
View Journal | View Issue

Cite this: *Nanoscale*, 2015, **7**, 9793

Automated quantitative image analysis of nanoparticle assembly†

Chaitanya R. Murthy, Bo Gao, Andrea R. Tao and Gaurav Arya*

Received 4th February 2015,
Accepted 20th April 2015

DOI: 10.1039/c5nr00809c

www.rsc.org/nanoscale

The ability to characterize higher-order structures formed by nanoparticle (NP) assembly is critical for predicting and engineering the properties of advanced nanocomposite materials. Here we develop a quantitative image analysis software to characterize key structural properties of NP clusters from experimental images of nanocomposites. This analysis can be carried out on images captured at intermittent times during assembly to monitor the time evolution of NP clusters in a highly automated manner. The software outputs averages and distributions in the size, radius of gyration, fractal dimension, backbone length, end-to-end distance, anisotropic ratio, and aspect ratio of NP clusters as a function of time along with bootstrapped error bounds for all calculated properties. The polydispersity in the NP building blocks and biases in the sampling of NP clusters are accounted for through the use of probabilistic weights. This software, named Particle Image Characterization Tool (PICT), has been made publicly available and could be an invaluable resource for researchers studying NP assembly. To demonstrate its practical utility, we used PICT to analyze scanning electron microscopy images taken during the assembly of surface-functionalized metal NPs of differing shapes and sizes within a polymer matrix. PICT is used to characterize and analyze the morphology of NP clusters, providing quantitative information that can be used to elucidate the physical mechanisms governing NP assembly.

1. Introduction

A significant challenge in nanoscience and emerging nanotechnologies is the ability to guide nanoparticle (NP) self-assembly with precise structural and functional control. Research efforts to control the spatial distribution of NPs spans a vast range of nanomaterials, physical properties, and applications. For example, diffuse networks of C_{60} and carbon nanotubes reinforce the mechanical stability of polymer films to prevent damage or dewetting,[1] and percolating networks of CdSe NPs serve as paths for electron transport within the active layer of bulk heterojunction photovoltaic cells.[2] We and others have pursued the hierarchical assembly of plasmonic metal NP clusters and strings for novel electromagnetic materials that serve to confine light.[3,4] In all of the above cases, NP assembly is highly dependent on interfacial and interparticle interactions, both of which dictate assembly events such as nucleation, growth, and coarsening. NP assembly is a highly dynamic process in which NPs aggregate to form larger clusters whose morphologies evolve with time. The ability to characterize these evolving NP cluster morphologies has important consequences for not only understanding the assembly mechanisms at work, but also for learning how to program these mechanisms for achieving desired NP cluster architectures.

The use of image analysis to characterize particle assemblies is a well-established practice in materials science. As early as in 1979, Forrest and Witten Jr.[5] used image analysis of transmission electron micrographs to identify and study the fractal structure of smoke particle aggregates.[5] Since then, image analysis has been used to characterize colloidal and NP aggregates in terms of various properties. For example, the size distribution of such aggregates has been analyzed by indigenously developed image analysis procedures[6–8] or *via* public-domain image-processing software.[9] While several of the public image analysis software, such as ImageJ,[10] are able to perform the kind of particle identification and counting required for calculating NP aggregate sizes, these software are generally not amenable to calculating other properties specific to NP assemblies. In particular, NPs are known to assemble into complex, higher-order structures such as strings, fractal trees, sheets, and percolating networks that require more elaborate metrics for characterizing their shape, self-similarity,

Department of NanoEngineering, University of California, San Diego, 9500 Gilman Drive, Mail Code 0448, La Jolla, CA 92093, USA. E-mail: garya@ucsd.edu; Fax: +858-534-9553; Tel: +858-822-5542

† Electronic supplementary information (ESI) available: Derivation and implementation of unbiased feature measurement, calculation of empirical distribution of single particle areas, calculation of self-similarity dimensions by regression on cluster data, and validation of image analysis algorithms. See DOI: 10.1039/c5nr00809c

and topology. While some studies have analyzed the fractal dimension[8,11] and shape metrics[12,13] of NP aggregates through image analysis, the related algorithms were not made publicly available as part of a software package, and, in many cases, the procedures were not automated. Thus, to the best of our knowledge, there does not exist at present a public-domain image analysis tool specifically tailored towards NP assembly.

Here we develop an automated image analysis tool capable of extracting in a high throughput manner useful data on NP assembly from images representing static snapshots of the material sample taken during the assembly process. Our Particle Image Characterization Tool (PICT) is coded in MATLAB R2012b and is freely available at the MATLAB File Exchange server.[14] PICT allows users to carry out various statistical analyses regarding the size, shape, and morphology of NP clusters for a given assembly time interval. It implements a novel algorithm to account for the variability in the NP building block and for biases in cluster sampling. To demonstrate the power of such a tool in analyzing NP assembly, we apply PICT to a series of assembly experiments that look at the evolution of plasmonic NP-polymer composites composed of metal NPs with different sizes and shapes. Previously, these NPs have been observed to assemble into structures ranging from interconnected NP networks to one-dimensional NP strings to NP clusters. By analyzing scanning electron microscope (SEM) images captured during the assembly process, we demonstrate how PICT can be used to characterize NP assembly morphologies and to determine the underlying assembly mechanisms.

2. Software development

2.1. Overview

The automated image analysis software developed here is coded in MATLAB R2012b and takes advantage of several functions present in MATLAB's Image Analysis Toolbox, in addition to our own algorithms. These are described or noted in the following sections wherever appropriate. All MATLAB functions, both existing and those developed here, are indicated in *typewriter font*. The software is designed with extendability in mind, and thus we attempted to maintain a modular structure throughout. The overall structure is summarized in Fig. 1. The software is split into five MATLAB files, each of which performs specific tasks:

- *PICT_import_images* imports and preprocesses the SEM images, preparing them for further analysis.
- *PICT_calibrate* lets the user calibrate our particle detection algorithm *via* a graphical user interface.
- *PICT_analyze_images* identifies particles and clusters, based on the calibrated settings, and measures cluster properties.
- *PICT_get_stats* combines data from all images and calculates bias-corrected size distributions and mean values for particle- and cluster-based properties along with bootstrapped error bounds for all calculated properties.

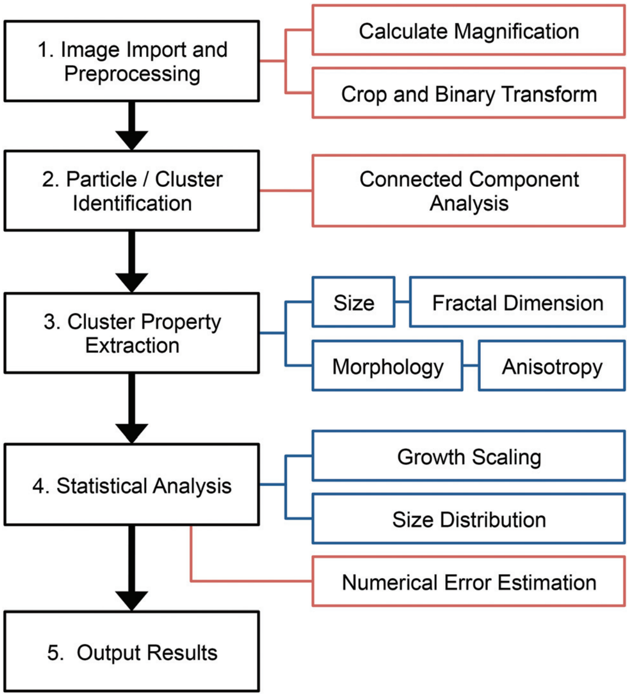
1. Image Import and Preprocessing
Calculate Magnification
Crop and Binary Transform
2. Particle / Cluster Identification
Connected Component Analysis
3. Cluster Property Extraction
Size
Fractal Dimension
Morphology
Anisotropy
4. Statistical Analysis
Growth Scaling
Size Distribution
Numerical Error Estimation
5. Output Results

Fig. 1 Flowchart summarizing our image analysis algorithm. Blue boxes contain some of the cluster properties and statistics calculated by the program. Red boxes contain some of the more general processes implemented.

- *PICT_export_results* contains routines to create and save a variety of images and figures for visualization of the results.

In each stage, the software organizes data into suitably named nested data structures that are saved to *.mat* files in the specified output directory and that are loaded by subsequent stages as needed. This ensures that all data can be loaded into MATLAB and accessed directly by the user at any time and should enable easy integration with external functions.

The software is straightforward to use. The user collects all images taken during an experiment into a single directory. The images need to be named according to a simple convention, which encodes essential information used by the software. The user then sets a few parameters, specifies the input and output path, and runs the five functions in turn. Further human interaction is required only during the calibration step. We have prepared a detailed user's guide, included with the software distribution as a PDF file, describing how to install and use the tool. This guide also includes the file-naming convention and complete documentation of all parameters, data structures, and exportable figures related to our software.

Below we describe how the software preprocesses the images, identifies individual NPs and their clusters, accumulates weights to correct for biases in cluster sampling and for polydispersity in NP sizes, and calculates various cluster properties.

2.2. Preprocessing of images

The steps involved in the preprocessing of SEM images are summarized in Fig. 2. In the first step, the software imports the raw grayscale images (Fig. 2a) and the image data is extracted and stored. SEM images typically have a black bar along the bottom that contains metadata in white (Fig. 2b). A high-threshold binary transform (*im2bw*) followed by morphological opening (*imopen*) with a rectangular structuring element (*strel*) removes all features from the image except for the rectangular scale bar (Fig. 2c and d). By measuring the length of the scale bar in pixels (the length in real units is encoded in the image filename), the precise magnification of the image is obtained. The program later uses the magnification of each image to convert all pixel-based measurements, such as cluster areas and lengths, into real units.

The original image is then cropped to the field of view (Fig. 2e). The user may specify the height of the metadata bar, in which case those rows of the image are simply removed. If the user does not provide this information, the program identifies the field of view itself by searching for the topmost row of the image that consists entirely of black pixels. The cropped image is then converted to binary (Fig. 2f); the threshold for this final binary conversion is calculated using MATLAB's *graythresh* function. Users with some programming experience should be able to modify the code to accommodate images from SEMs that store metadata in a different format or location.

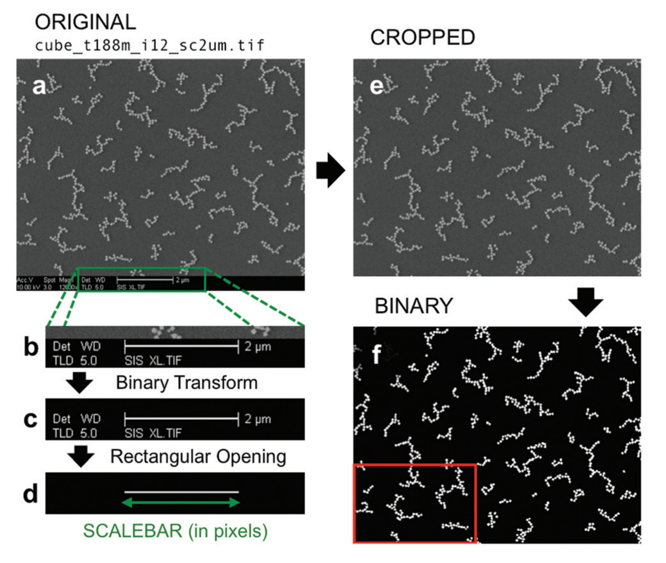

Fig. 2 Preprocessing of images. Images show actual data from program execution. (a) An original unaltered SEM image. The filename listed above contains formatted information about the particle type, the experiment time in minutes, the image number for that time point, and scale bar length in microns. (b–d) Procedure for isolating and measuring the scale bar. (e) Cropping of the image to the field-of-view and (f) conversion to a binary image. The region within the red box is used later in Fig. 3 to illustrate identification of particles and clusters.

The fraction of pixels that are part of the foreground in the binary image gives the total area fraction Φ of the NPs. Note that Φ is related to the total projection area of all particles, independent of whether they are arranged in clusters or not. Φ may vary slightly from one image to the next, but its average value should remain approximately constant as a function of time for a given experiment. Large fluctuations or systematic trends in $\Phi(t)$, where t is the time, may indicate inadequate or biased imaging of the sample.

2.3. Identification of particles and clusters

The procedure used to identify particles and clusters is illustrated in Fig. 3. Throughout this paper, we use the term "particle" to refer to the individual primary NPs that assemble to form larger structures and the term "cluster" is used to refer to an assembled structure consisting of one or more NPs. The software identifies all "objects" (continuous foreground regions) in the binary image by image segmentation (*bwconncomp*). All objects that are touching a border are removed (*imclearborder*), and the remaining objects are characterized using *regionprops*.

Single particles are identified on the basis of three properties of the objects: area, solidity, and eccentricity. The area of an object is calculated as the number of foreground pixels occupying it, appropriately converted to real units. The solidity of an object is calculated as the ratio of its area to the area of the smallest convex polygon (convex hull) completely enclosing that object. Finally, its eccentricity is calculated as the ratio of the distance between the foci of an ellipse that has the same normalized second central moments as the object and the major axis length of the ellipse. Since NPs are usually convex polyhedra, the solidity for a single particle will typically be greater than for a dimer, trimer, or larger aggregate. Eccentricity, on the other hand, varies widely with NP species—low for spheres and cubes; high for rods. The acceptable range for each property is specified by the user during calibration. As an example, property ranges used in the analysis of images from three of our experiments are given in Table 1. Finally, particle data from all images is combined to calculate distribution functions for particle areas and for various shape properties. Common measures such as the effective particle diameter and the polydispersity index are either computed and saved or are readily obtainable (in the case of measures that PICT does not explicitly calculate) from these distribution functions. For a detailed list of all computed quantities, along with their precise definitions, we refer readers to the documentation included with PICT. The particle identification method implemented in our program is similar to the one used in the Analyze Particles tool in the popular ImageJ software.[10]

To identify clusters, we first note that particles within a cluster may sometimes be separated from one another by small gaps. The whole collection of particles should still be properly identified as a single cluster, rather than as two or more smaller chains, as would occur if one naively found connected components in the original binary image (as in Fig. 3b). To achieve this, we implement a procedure for

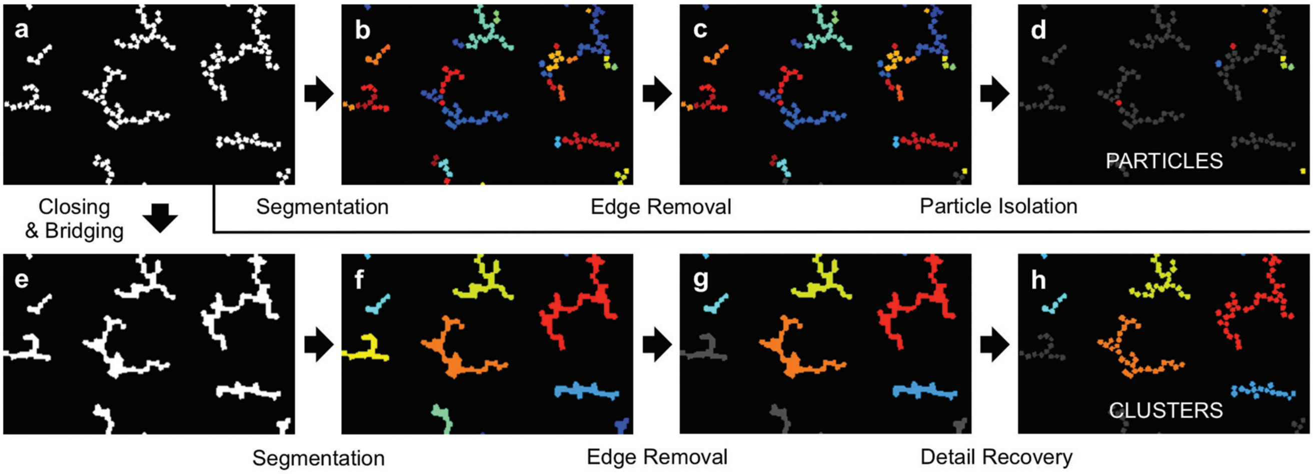

Fig. 3 Schematic for identification of particles and clusters. Images show actual data from program execution. (a) Close-up of the lower-left corner of the SEM image shown in Fig. 2; the precise region is marked in red in Fig. 2f. (b–d) Steps in the identification of particles. As this SEM image was taken late in the assembly process, only a few particles were identified. Images taken at earlier times naturally yield many more particles. (e–h) Steps in the identification of clusters.

Table 1 Property ranges used in identification of particles

Species	Area (nm^2)	Solidity	Eccentricity
Cubes	2000–12 000	0.8–1.0	0–0.7
Spheres[a]	200–2000	0.85–1.0	0–0.7
Rods	200–1200	0.8–1.0	0.9–1.0

[a] ~30 nm diameter in 28k PS matrix (see Methods).

"almost-connected-component labelling" described by Eddins.[15] Starting again from the binary image, the software performs a morphological "closing" (*imclose*) with a small structuring element. This joins any particles that are not quite touching, yet are close enough to be considered part of a cluster. The user may either specify the gap below which particles are joined in this manner, or the software will attempt to determine it automatically based on measured particle sizes. Next, each connected foreground region is identified (*bwconncomp*) and labeled (*labelmatrix*), and regions that extend beyond the field of view are removed. This removal introduces a bias, since larger clusters are more likely to extend beyond the edges of the image than are smaller clusters; this bias is corrected by attaching weights to each observed cluster, as explained in the next section. Finally, a logical AND operation between the "dilated image" and the original binary image recovers any fine details of the cluster shape.

2.4. Unbiased feature measurement

Objects that appear to extend beyond the edges of an image cannot be measured because their size and shape outside the image area is unknown. If only objects that are fully contained within the image are considered, however, the proportion of large and small objects will be inaccurately computed, since it is more likely for larger objects to touch an edge. This bias may be compensated for by attaching size-dependent weights to each object that is measured.[16]

To estimate these weights, we assume that for each object measured in the image, a large number of identical objects exist in the sample, with equal fractions oriented in any given direction. Imagine that we randomly pick a field of view that contains the centroid of one of these objects. If the centroid is located far enough from the edges of the image, the entire object will fit within the field of view and will be measurable. The conditional probability of this happening is simply the area of the region where the centroid can be located so that the object is fully contained, relative to the total area of the image:

$$P = \frac{(W_x - F_x)(W_y - F_y)}{W_x W_y} \qquad (1)$$

where W_x and W_y are the dimensions of the image in the x and y directions, and F_x and F_y are the maximum dimensions of the object in those directions.[16]

If n_i objects of type i exist per unit area of the sample, we would expect to fully observe only $n_i P_i$ of them per unit area that we image. Therefore, we attach a weight of $w_i = 1/P_i$ to each measured object. In principle, this recovers an unbiased estimate of the true sample statistics. The objects are oriented at random, however, so eqn (1) must first be averaged over all possible orientations θ. The bias-correction weights are then given by

$$w_i = \frac{1}{\langle P_{i\theta} \rangle_\theta} = \frac{W_x W_y}{\left\langle (W_x - F_\theta)\left(W_y - F_{\theta+\frac{\pi}{2}}\right) \right\rangle_{\theta \in [0,\pi)}} \qquad (2)$$

where F_θ is the extent of the object along the θ-direction, angle brackets denote averaging, and only angles smaller than

π radians need to be considered in the average because $F_{\theta+\pi} = F_\theta$. When calculating statistics, the program assigns each measured cluster a weight according to eqn (2). The precise formula that we used for computing w_i in the software and its derivation are provided in the ESI.†

2.5. Calculation of cluster size distribution

The cluster size distribution $N_s(t)$, the number of s-particle clusters observed at time t, is a fundamental quantity in all assembly experiments. The details of this distribution can often reveal a great deal about the underlying assembly process. Size distributions at different time points or from different experiments can only be meaningfully compared with one another after they have been normalized. Our software computes the relative size distribution

$$\nu_s(t) = \frac{N_s(t)}{\sum_s sN_s(t)} \tag{3}$$

Theoretical and simulation results[17,18] frequently employ the number concentration $n_s(t)$, the number of clusters of size s per unit area. This is related to the quantity defined above *via* $n_s(t) = \nu_s(t)\Phi(t)/\bar{a}$, where $\Phi(t)$ is the total occupied area fraction and $\bar{a}$ is the mean particle area.

The cluster size distribution may also be characterized *via* its moments. In particular, the number-average cluster size

$$S_1(t) = \frac{\sum_s sN_s(t)}{\sum_s N_s(t)} \tag{4}$$

and "mass-average" cluster size

$$S_2(t) = \frac{\sum_s s^2N_s(t)}{\sum_s sN_s(t)} \tag{5}$$

are simple measures of the overall assembly rate, and are both widely used in the literature.[19] The ratio between $S_2(t)$ and $S_1(t)$ provides concise qualitative information about the shape of the distribution, and thereby about the assembly process.[20] Note that the interpretation of $S_2(t)$ as the mass-average cluster size is rigorously valid only if we neglect NP polydispersity; otherwise $S_2(t)$ may be thought of as a particular quantity, defined in eqn (5), that characterizes the cluster size distribution.

If a human were tasked with finding $N_s(t)$, they might simply count by eye the number of particles in each cluster and tally the results. This procedure of "direct counting" is difficult to automate, as it requires the software to differentiate features (particles within clusters) that are very closely spaced and whose boundaries are not crisp. In the interest of simplicity and robustness, we calculate cluster sizes indirectly from area measurements. Specifically, we obtain the best possible estimate of $N_s(t)$ from the measured area A_k of cluster k and measured probability density $p(a)$ of single-particle areas. Both types of data are readily obtained by counting the number of pixels in each identified cluster or single particle and converting the measured pixels to square nanometers using the image scale bar.

In one approach, the size of a given cluster may be estimated as

$$s_k = \frac{A_k}{\bar{a}} \tag{6}$$

where $\bar{a} = \int ap(a)da$ is the mean single-particle area. To obtain $N_s(t)$, one could round each of these sizes to the nearest integer and accumulate a histogram. While intuitively appealing, this method is not theoretically sound. It entirely ignores the polydispersity of the particles—if cluster k happens to contain many particles that are smaller or larger than average, eqn (6) would under- or over-predict s_k—and the rounding further harms the statistical reliability of the result.

A much better approach is to apply Bayesian inference.[21] Consider the conditional probability $P(s|A)$ that a cluster with measured area A actually contains s particles. Using Bayes' theorem and the law of total probability,

$$P(s|A) = \frac{P(A|s)P(s)}{P(A)} = \frac{P(A|s)P(s)}{\sum_{s'} P(A|s')P(s')} \tag{7}$$

where $P(A|s) \equiv p(A|s)dA$ is the probability that an s-particle cluster will have an area infinitesimally different from A. Before we measure the area of a cluster, we have no information about how many particles it contains. It is therefore appropriate to use a uniform prior, $P(s) = P(s')$. Consequently, these factors cancel, and eqn (7) simplifies to

$$P(s|A) = \frac{p(A|s)}{\sum_{s'} p(A|s')} \tag{8}$$

The calculation of $p(A|s)$ is straightforward. Empirically, we find that the distribution of single-particle areas is approximately Gaussian with mean $\bar{a}$ and variance ${\sigma_a}^2$ (see ESI†). However, if $X_j \sim \mathcal{N}(\mu, \sigma^2)$ are n independent, identically distributed Gaussian random variables, then their sum $Z = \sum_{j=1}^{n} X_j$ is also normally distributed: $Z \sim \mathcal{N}(n\mu, n\sigma^2)$. Applying this result to our scenario, we find that

$$p(A|s) = \frac{1}{\sqrt{2\pi s}\,\sigma_a} \exp\left(-\frac{(A - s\bar{a})^2}{2s{\sigma_a}^2}\right) \tag{9}$$

In principle, eqn (8) together with eqn (9) allow us to construct the best possible estimate of the true cluster size distribution $N_s(t)$, given our area measurements. The overall algorithm may be summarized as follows:

- Identify all single particles and compute $\bar{a}$ and ${\sigma_a}^2$.
- For each distinct cluster j found,
  1. Measure its area A_j.
  2. Calculate the bias-correcting weighting factor w_j, as described previously.

3. Update the size distribution according to

$$N_s(t_j) = N_s(t_j) + \frac{p(A_j|s)}{\sum_{s'} p(A_j|s')} w_j$$

with $p(A_j|s)$ and $p(A_j|s')$ given by eqn (9).

- Normalize $N_s(t)$ according to eqn (3) to obtain $\nu_s(t)$.

Our software implements the above algorithm with slight modifications to improve code performance. Fig. 4 compares the cluster size distribution obtained without bias correction or Bayesian estimation (Fig. 4a) against the corresponding distributions obtained with bias correction (Fig. 4b) and with both bias correction and Bayesian estimation (Fig. 4c). The above distributions were all obtained at one representative time point during the assembly of 80 nm Ag nanocubes (see Methods).

It is sometimes the case that the size distribution of primary NPs in an experiment is better described by a log-normal distribution (or some other distribution) than by a Gaussian. Our Bayesian algorithm may be generalized to handle an arbitrary probability distribution $p(a)$ of single-particle areas; one simply needs to replace eqn (9) for $p(A|s)$ by the appropriate distribution for $A = \sum_{j=1}^{s} a_j$, where the a_j are independent and identically distributed according to $p(a)$. This $p(A|s)$ may be computed by standard methods described in books on probability theory or statistics.

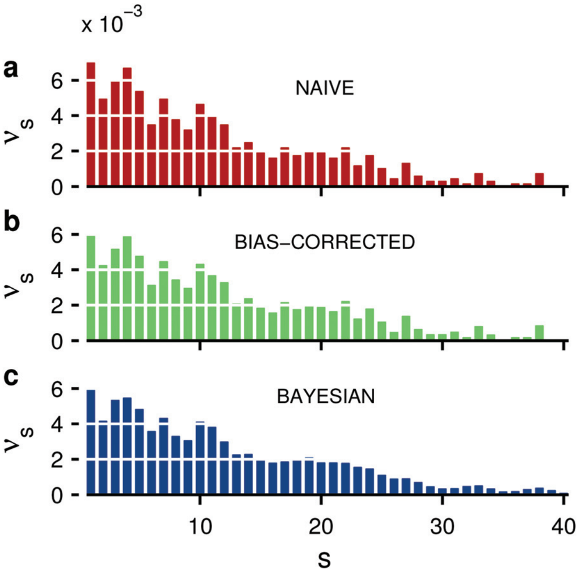

Fig. 4 Comparison of normalized size distributions obtained by different methods. (a) Simple accumulation of cluster sizes computed with eqn (6). (b) Same, except that each cluster is weighted according to eqn (2) when constructing the histogram. (c) Accumulation by Bayesian inference (also bias-corrected). Notice that bias-correction decreases the relative count of smaller particles. The Bayesian distribution is visibly smoother due to elimination of statistical artifacts present in the other methods.

2.6. Calculation of cluster dimensions and anisotropy

Our software calculates the eigenvalues (principal moments) ${R_1}^2 \geq {R_2}^2$ of the gyration tensor for each cluster (see Fig. 5b) by using MATLAB's *regionprops* function. The eigenvalues are then used to compute the radius of gyration of the clusters, which characterizes their spatial extent, and various other useful properties, such as aspect ratio, anisotropy ratio, elongation, and fractional anisotropy, which all characterize the anisotropy of clusters in different ways:

1. Radius of gyration

$$R_g = \sqrt{R_1^2 + R_2^2} \tag{10}$$

2. Aspect ratio

$$\mathrm{AR} = \frac{R_1}{R_2} \tag{11}$$

3. Anisotropy ratios, as defined by Botet and Jullien[22]

$$A_1 = \frac{\langle {R_1}^2 \rangle}{\langle {R_2}^2 \rangle} \quad \text{and} \quad A_1' = \frac{\langle {R_1}^2 \rangle}{\langle {R_2}^2 \rangle} \tag{12}$$

Here, angle brackets denote averages over a set of clusters. Note that A_1' is the mean square aspect ratio.

4. Elongation, also known as Relative Anisotropy[23]

$$\varepsilon = \frac{{R_1}^2 - {R_2}^2}{{R_1}^2 + {R_2}^2} \tag{13}$$

5. Fractional Anisotropy[23]

$$\kappa = \frac{{R_1}^2 - {R_2}^2}{\sqrt{{R_1}^4 + {R_2}^4}} \tag{14}$$

2.7. Calculation of cluster backbone length and end-to-end distance

Our results show that NPs tend to assemble into fractal chains resembling branched polymers.[3] By analogy with branched

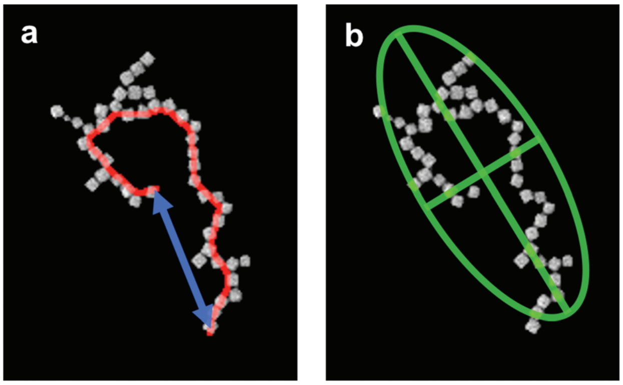

Fig. 5 Representative NP cluster depicting: (a) backbone path (red) and end-to-end distance vector (blue), and (b) fitted ellipse with major and minor axes (green). The major and minor radii have length $2R_1$ and $2R_2$ respectively.

polymers, we can treat such clusters as having a central backbone of length L from which other shorter chains branch off. The Euclidean distance between the two ends of this backbone define the end-to-end distance L_{ee}.

A connected region can be viewed as a graph where each foreground pixel is a "vertex", and any vertices corresponding to adjacent pixels are connected by an "edge". The "geodesic distance" between two vertices is the length of the shortest path between them (the minimum number of edges needed to get from one to the other). In this context, the backbone length L is called the "graph diameter", and it is defined as the maximum over the set of all pairwise geodesic distances. In practice, however, computing the maximum over the set of shortest paths is a very time-consuming process, especially for large clusters. We abandon rigor in favor of speed in this case, and employ the following algorithm:

1. Identify all pixels (vertices) $\{\nu\}$ on the boundary of a cluster.
2. Pick one of these, ν_0, at random.
3. Find the vertex ν_1 that is furthest from ν_0 (in the geodesic sense).
4. Find the vertex ν_2 that is furthest from ν_1 (in the geodesic sense).
5. Assume that ν_1 and ν_2 are the "peripheral vertices" – the geodesic between them is the backbone and the Euclidean distance between them is the end-to-end distance L_{ee} (see Fig. 5a).

In principle, the algorithm presented above can fail for certain pathological graphs and choices of ν_0. However, the chances of encountering such graphs in the binary images of experimentally observed fractal aggregates is negligible. We refer interested readers to books on graph theory (in particular, on algorithms for computing the graph diameter) for more details.

2.8. Calculation of cluster self-similarity dimensions

We find that NP clusters have structures that are self-similar over a range of length-scales.[24] The nature of this self-similarity can be most naturally described in terms of various fractal or scaling dimensions $\{D\}$.[25] The values of $\{D\}$ can give insight into the mechanisms by which the clusters assemble. Indeed, many models of colloidal aggregation have been studied over the years, and different models often yield aggregates with characteristic self-similarity dimensions.[26]

2.8.1. Calculation for a family of clusters. We can calculate the self-similarity dimensions of a self-similar object in several ways.[24,27] One way is to determine how the object's mass scales with some linear dimension associated with the object (its radius of gyration, longest axis, *etc.*); if l denotes the linear measure and m the mass, we should find that

$$m \sim l^{\mathrm{D}} \tag{15}$$

To use eqn (15), we must either track one self-similar object as it grows or shrinks, or else study a family of similar but differently-sized objects to extract the scaling exponent. The former method is only applicable if the growth process can be followed in real time. Our image analysis software therefore implements the second method: self-similarity dimensions are estimated by regression on large sets of cluster data.

The software computes D as the slope of a straight-line fit to log(s) *versus* log(l) data, where s is the estimated size of a cluster (as defined in eqn (6)). This method was found, based on an analysis of the distribution of regression residuals,[28] to be superior to fitting a power-law directly to s *versus* l (see ESI† for more details). A lower cutoff size is enforced for clusters included in the regression, to ensure that polydispersity in individual particle sizes does not contaminate the results. Each distinct length scale l yields a corresponding exponent D. Two particularly important examples are the radius of gyration R_{g}, which yields a fractal dimension that we denote as $D_{\text{f}}^{(\text{R})}$:

$$s \sim {R_{\text{g}}}^{D_{\text{f}}^{(\text{R})}} \tag{16}$$

and the graph diameter or backbone length L, which yields the so-called "chemical" (or "spreading") dimension d_{L}:[27]

$$s \sim L^{\mathrm{d_L}} \tag{17}$$

A plot depicting the calculation of $D_{\text{f}}^{(\text{R})}$ is shown in Fig. 6a. Other computed dimensions are described in the documentation included with our software.

2.8.2. Calculation for individual clusters. Another method for computing the fractal dimension of an object involves calculating its radial cumulative distribution function $C(r)$.[24] Essentially, $C(r)$ is the fraction of space occupied by the object within a circle of radius r centered at any reference point in the object. For a fractal, this function should decay as

$$C(r) \sim r^{\left(D_{\text{f}}^{(\text{C})}-d\right)} \tag{18}$$

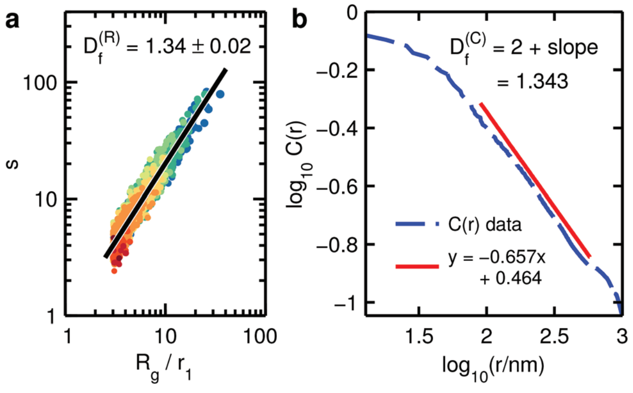

Fig. 6 (a) Log–log plot of cluster size s *versus* normalized radius of gyration R_{g}/r_1. Data points represent all clusters observed during the experiment labelled E1, colored by time of observation (see Fig. 7a–d). r_1 is the mean single-particle radius of gyration. The best-fit line is shown and the fractal dimension $D_{\text{f}}^{(\text{R})}$ given by the slope of this line is listed; error bounds indicate 95% confidence interval for the slope. (b) Plot showing the decay of $C(r)$ with distance r for the NP cluster shown in Fig. 5, and the computed fractal dimension $D_{\text{f}}^{(\text{C})}$.

where d is the dimension of the space in which the fractal is embedded. The r-domain over which this scaling relation holds is exactly the range of length-scales over which the object is self-similar. Note that we explicitly distinguish the fractal dimension $D_f^{(C)}$ entering into the above relation from the dimension $D_f^{(R)}$ described previously, because the values are in general different for finite objects or for objects whose structure is only approximately fractal, such as NP clusters.

In a binary (thresholded) digital image of a two-dimensional NP, each cluster is a set of pixels. These pixels are points on a lattice and their coordinates are pairs (x_i,y_i) of integers. Let $f(r)$ be the number of arbitrary pairs (x,y) of integers for which $x^2 + y^2$ does not exceed r. A circle of radius r drawn around any reference pixel (x_k,y_k) will contain $f(r)$ lattice points, but only $N_k(r)$ of these will be occupied by the aggregate. The radial cumulative distribution function $C(r)$ is then given by

$$C(r) = \frac{\langle N_k(r) \rangle_k}{f(r)} \tag{19}$$

where we have averaged over reference points. The average must be defined with care; if the aggregate is finite, all reference points are not equivalent. That is, a given point can only contribute to $C(r)$ over the interval $0 \le r < R_k$, where R_k is the radius of the largest circle that can be drawn around (x_k,y_k) without extending beyond the edges of the finite aggregate. To determine this upper limit explicitly, we first calculate the distance of each point in the aggregate from its centre of mass:

$$r_k^c = \sqrt{(x_k - \langle x_i \rangle)^2 + (x_k - \langle y_i \rangle)^2} \tag{20}$$

A circle of radius $\{r_k^c\}$ drawn around $(\langle x_i \rangle, \langle y_i \rangle)$ would just contain the whole aggregate, and we use this circle as our boundary. Therefore, we have

$$R_k = \max\{r_k^c\} - r_k^c \tag{21}$$

With this restriction on the domain of r to which each reference point k can contribute, we construct $C(r)$ by applying eqn (19). Then, we can extract the fractal dimension of the aggregate from a plot of log(C) *versus* log(r), in accordance with eqn (18). Empirically, we find that such a plot is linear for $d_1 < r < R_g$, where $d_1 = \left\langle \sqrt{4a/\pi} \right\rangle$ is the mean "equivalent diameter" of a single nanoparticle (a is the single-particle area) and R_g is the radius of gyration of the aggregate. A representative plot is shown in Fig. 6b.

2.9. Statistics and error analysis

The entire set of statistics calculated by our software, including those mentioned above, are described in the documentation distributed with our code. The software also computes and returns standard errors and confidence intervals for most calculated quantities. These are obtained by the nonparametric bootstrap method,[29] which allows for easy and reliable uncertainty estimation for statistics, such as weighted means and ratios, for which simple analytical error expressions are not available.[30]

2.10. Software validation

We have carried out various tests to validate the three main components of our software: the algorithms for detecting individual particles and their clusters; the calculation of various structural properties of the particle clusters; and the Bayesian algorithm for the estimation of cluster size distributions. A detailed discussion of all three forms of validation is provided in the ESI.†

3. Results and discussion

To demonstrate the applicability of our image analysis tool, we have used it to investigate similarities and differences in the assembly of NPs carried out with particles of different sizes and shapes within polymer films of different molecular weights. Specifically, we analyzed five different experiments involving the assembly of: silver nanocubes of edge length 80 nm in polystyrene (PS) thin films of molecular weight (M_w) ≈ 11 000 (11k) (we refer to this experiment as E1); gold nanorods of length 40 nm and diameter 10 nm (E2) and Gold nanospheres of diameter 13 nm in PS films of the same M_w (E3); and silver nanospheres of diameter 30 nm in PS films with shorter chains of M_w ≈ 3000 (E4) and with longer chains of M_w ≈ 28 000 (E5). The NPs were synthesized using wet chemical methods to yield a colloidal dispersion of uniformly sized and shaped particles surface-functionalized with poly(vinyl pyrrolidone) (PVP) chains of M_w ≈ 55 000. The NP dispersion was spread onto an air-water interface to produce a monolayer of uniformly distributed NPs, which was then transferred onto the surface of a thin PS film. The NPs were embedded into the underlying polymer through solvent annealing, whereupon the NPs sank into the film and began to diffuse within it and assemble into higher-order NP clusters. In each experiment, we captured on the order of 100 SEM images, spread across all time points; Fig. 7a shows smaller sections of a representative SEM image captured from each of the five experiments. Details on the synthesis, assembly, and imaging of NPs are provided in Methods.

Fig. 7b–g summarizes the results from our analyses of the above five NP assembly experiments. The total NP area fraction Φ calculated for each SEM image that was analyzed is plotted in Fig. 7b. As discussed earlier, Φ is calculated as the fraction of foreground pixels in a thresholded image. On the whole, the mean area fraction remains largely constant as a function of time for each experiment, as it should. Variations in Φ on the order of 1–2% between images may be explained as the result of local inhomogeneities induced by the stochastic motion of the particles and clusters. The few outliers are likely the result of slightly nonuniform sample preparation. Also, as mentioned in Methods, the data for different time points must come from different samples due to the requirement of the polymer samples to be frozen during imaging. Note that the area fractions in our experiments are high enough that shape effects are significant, and mean-field theories, such as the

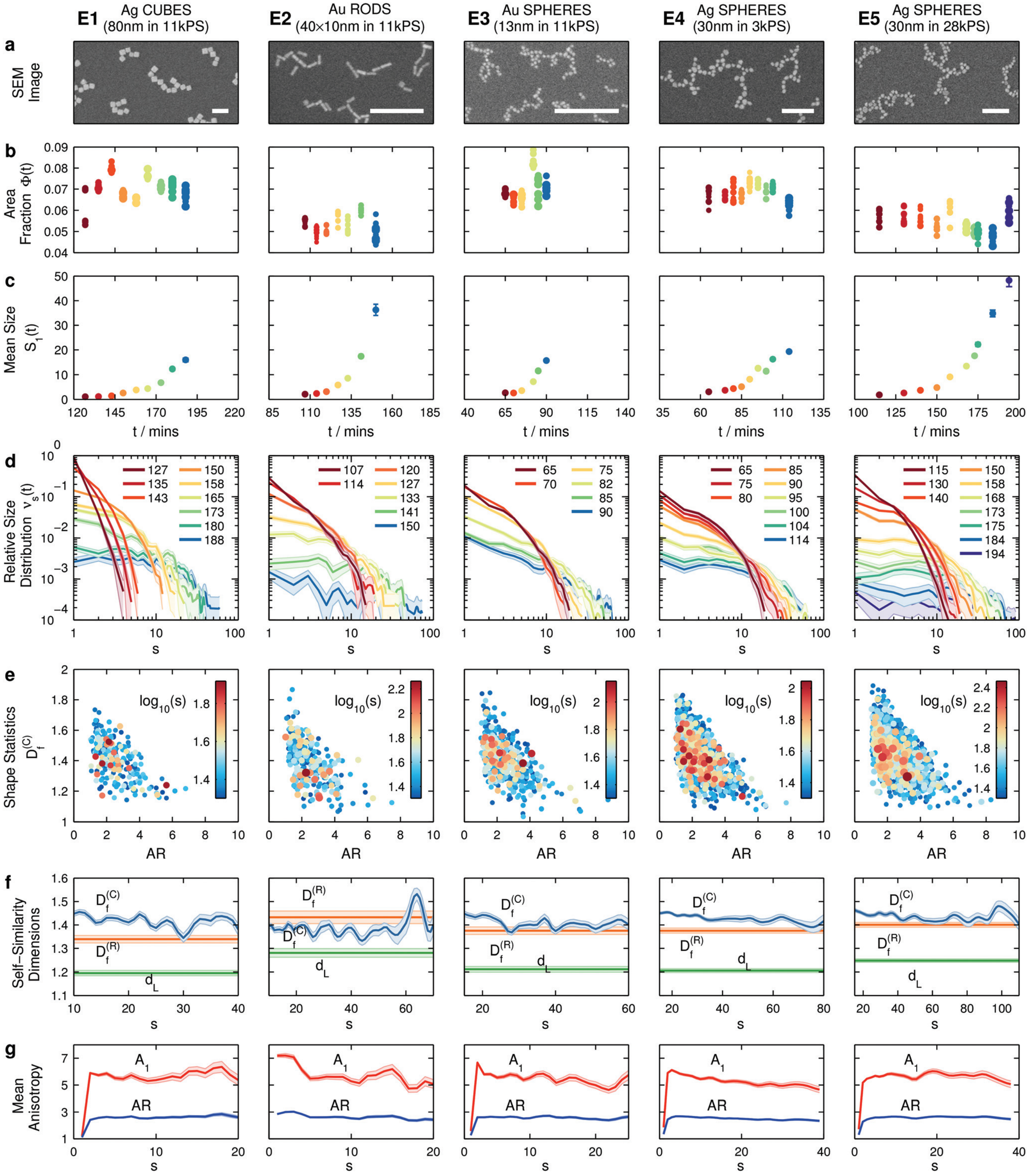

Fig. 7 Results from automated image analysis of data from five sets of experiments organized in columns. (a) Representative sections of analyzed SEM images. Scalebars are 250 nm. (b) Scatterplot of particle number density in each analyzed image as function of time. The points are sized relative to image area. (c) Mean cluster size versus time in minutes. Axes scales are identical to aid visual comparison of the growth rates. (d) Normalized cluster size distributions. (e) Scatterplot of fractal dimension versus aspect ratio. Points correspond to individual clusters, and are sized/colored relative to cluster size. (f) Mean fractal dimension $D_f^{(C)}$ versus cluster size. Overall values obtained for $D_f^{(R)}$ and chemical dimension d_L are also plotted. (g) Anisotropy ratio A_1 and mean aspect ratio AR versus cluster size.

classical Smoluchowski model of particle aggregation,[17] may not apply.

The growth rates of the number-average mean cluster size $S_1(t)$ with time are plotted in Fig. 7c. In all experiments, chloroform vapor annealing of the polymer-NP films begins at $t = 0$, but we observe a significant dormant period preceding the onset of aggregation. Prior work has shown that the well-dispersed NPs, which are initially deposited on top of the polymer film, slowly sink into the film as it becomes mobile, until they are fully embedded by the polymer.[3] It is unclear why aggregation does not occur simultaneously during this process. It is possible that the grafted chains on the NP surface prevent immediate aggregation, or that the lateral diffusion coefficient of the particles slowly increases from zero as the polymer becomes mobile over larger length scales. The delayed start times for the larger particles (E1) and higher M_w polymer matrix (E5), are consistent with this theory, but the late start-time for the smaller nanorods (E2) is unusual.

The exponential shape of the growth curves is suggestive of reaction-limited aggregation kinetics,[31] but the uncertainty in start times makes it difficult to extract exponents, or to quantitatively compare rates between experiments. It is possible that diffusion-limited kinetics, conspiring with a time-dependent diffusion coefficient, could also lead to growth scalings of this kind. The overall growth rates, once aggregation begins, are comparable across experiments E1, E4 and E5. Nanorods (E2) and small 13-nm nanospheres (E3) seem to exhibit faster kinetics, but in the latter case, limited data makes it difficult to conclude this with certainty. Faster growth in these cases may be explained on the basis of faster diffusion of smaller particles through the polymer matrix, resulting in higher collision rates, and hence faster growth.

Cluster size distributions (Fig. 7d) are strikingly similar across all five experiments, exhibiting a characteristic monotonic decay of $\nu_s(t)$ with s at all times. The absence of a peak in the distributions suggests that the dominant mode of cluster growth at later times is *via* cluster-cluster, rather than particle-cluster collisions. If the latter were predominant, then we should expect to observe a depletion of single particles, which we do not. The similarity in shape of the distributions is somewhat remarkable, given that the aggregating particles differ widely in shape and size as well as in the properties of the surrounding matrix. Our observation of cluster-cluster collisions is also consistent with the step-growth polymerization mechanism reported by Liu *et al.*[32] for polymer-functionalized gold nanorods. It should be noted that at high enough NP densities, large clusters collide to form nearly interconnected networks, which can make cluster sizes difficult to determine accurately.

The scatterplots in Fig. 7e show shape data (fractal dimension $D_f^{(C)}$ *versus* aspect ratio AR) for all NP clusters larger than $s = 20$ observed in each experiment. The points are sized and colored according to the size of the corresponding cluster. The overall shape of the cloud of points is similar for all experiments and shows that there is large variation in shape between individual clusters. Low fractal dimensions and high aspect ratios suggest extended chain-like morphologies, while the opposite combination of properties suggests compact, globular shapes. Low D_f and AR together imply loose dendritic structures. In all five experiments, small clusters can be found that exhibit each of these morphologies; as the clusters grow larger, they tend to converge to fractal dimensions near 1.4 and aspect ratios near 2–3.

As noted earlier, self-similarity dimensions provide a concise and powerful characterization of the shape of aggregating clusters. The fractal dimension $D_f^{(R)}$, and the chemical (or spreading) dimension d_L calculated in each experiment are marked respectively as orange and green horizontal lines in Fig. 7f. These lines indicate the values obtained by regression on all clusters (larger than a lower cutoff size) observed during the respective experiments. The calculated dimensions are, within error bars, approximately equal across experiments E1 and E3–E5, but are lower than the values predicted by both diffusion-controlled Brownian aggregation models ($D_f \approx 1.44$) and reaction-controlled models ($D_f \approx 1.55$).[26] This lower dimensionality likely reflects the steric interactions between grafted polymer, which have been proposed to favor linear rather than branched configurations in clusters composed of polymer-grafted nanoparticles.[33] It is interesting that both dimensions are significantly higher for nanorods (E2) than for the other particle species studied. This somewhat counterintuitive result may be due to rods starting to align side-by-side in larger aggregates.

Less data is available in the literature for the chemical dimension d_L than for D_f. This dimension appears to have been studied mostly in the context of percolation clusters, a particular type of model that is not really applicable to our experiments. Nevertheless, we find that the ratio $d_{\min} = D_f/d_L$, computed from our results, matches the reported value for percolation clusters in two dimensions ($d_{\min} = 1.13$).[34] Further work is needed to determine whether there is a physical explanation behind this result or if it is merely a coincidence. The mean $D_f^{(C)}$ values as functions of cluster size are also plotted (in blue) in Fig. 7f. These appear to slowly converge to the corresponding $D_f^{(R)}$ values at large cluster sizes, as expected.

Fig. 7g shows anisotropy ratios A_1 and mean aspect ratios AR as functions of cluster size. These values start off large for nanorods (E2), and then equilibrate to smaller values as cluster size increases, while for the other particle species the opposite trend is observed. This merely reflects the inherent anisotropy of individual nanorods, as compared with the relative isotropy of nanocubes and nanospheres. The high mean anisotropy of the clusters in all five experiments rules out certain models, such as Witten and Sander's particle-cluster aggregation (PCA), which yield aggregates that are isotropic and dendritic on average.[35] However, the results match those for the cluster-cluster aggregation (CCA) model[36,37] extremely well. The anisotropy of clusters generated in different varieties of CCA was studied in detail by Botet and Julien,[22] who obtained $A_1 = 5.7 \pm 0.2$ in the diffusion-controlled version and $A_1 = 4.7 \pm 0.2$ in the reaction-limited case.[22] Our results are certainly in this range, and lean towards the former value.

However, it is also possible that reaction-limited kinetics, coupled with restructuring due to repulsive steric interactions between grafted polymer, could produce clusters with the observed anisotropies. Further studies are needed to decide the issue.

The results presented and discussed above represent only a subset of the data and statistics that our tool calculates. Other results may be useful for exploring specific questions about nanoparticle aggregation, and in such cases, similar plots can be easily generated. We have included routines for generating several of these plots in the *PICT_export_results* function of our software package. It should also be noted that our experiments were planned and performed before the image analysis tool was conceived. Thus, there is significant room for further improving the reliability of the data presented in Fig. 7 and of the conclusions derived from this data. For instance, in each of our experiments, we captured roughly one hundred SEM images, spread across all time points. This number is sufficient for analysis, but more images would yield even better statistics, without any significant increase in the time needed to process the additional data. In principle, the ideal method of capturing images would be to pick a large number of non-overlapping image locations at random, and have the microscope automatically image the sample at those coordinates. Such a method minimizes otherwise unavoidable biases on the part of the researcher searching for “good” spots to image.

4. Conclusions

We have introduced a quantitative image analysis software for studying the assembly of NPs. The software inputs a set of experimental electron microscopy images captured during assembly and outputs various structural properties of the NPs and their clusters as a function of time. In particular, the software computes both averages and distributions in various metrics characterizing the size, morphology, anisotropy, and self-similarity of NP assemblies along with error bounds in each computed property. To demonstrate its applicability, we have used the software to analyze particle assembly carried out with NPs of different shapes and sizes in polymer films of different molecular weights. Our results illustrate how the software can be used to not only characterize NP assembly but to also dissect the role of different assembly parameters and elucidate underlying mechanisms of NP assembly.

We envision that our image analysis software will become an invaluable tool for researchers studying NP assembly in various media; the software is already almost fully automated and user friendly. By minimizing the time and effort required for data analysis, our tool should enable researchers to more rapidly study many interesting problems in this field and to also collect larger, more statistically reliable datasets. In fact, other members of our group are already using the tool to explore some of these questions. Future applications that we plan to pursue include a systematic analysis of the effect of various system parameters on the growth and morphology of NP aggregates: parameters such as particle size, number density, and lengths of the grafted and matrix polymer. We also intend to perform a more detailed comparison of our experimental results with simulations and theoretical models of aggregation.

In our demonstration, we used PICT to analyze SEM images of noble metal NPs; however, PICT is not limited to characterization of only these images/materials. PICT can be used to analyze any image [e.g., SEM, transmission electron microscope (TEM), and optical] as long as the imaging contrast can be made high enough to resolve individual particles and that the particles are reasonably uniform in size and shape. For TEM images where NPs are supported on a mounted grid (typically backed with amorphous carbon), PICT may encounter limitations for analyzing NPs that possess limited contrast with the grid backing, namely small NPs (<5 nm) that are composed of low electron density materials such as polymers or oxides. In this limiting case, setting threshold contrast values too high may lead to the incorrect recognition of parts of grid background as small particles.

The software has room for further extension and improvement. For instance, it cannot at present handle multiple NP species in the same experiment, such as a binary mixture of nanospheres and nanocubes. However, our software code is written with extendability in mind, and it should be possible to add this functionality fairly easily. We also plan to incorporate other features into the tool in the future, including kernel-based detection of particles within clusters, correlation functions for nanoparticle or cluster orientations, and additional methods like box-counting for calculating self-similarity dimensions. We hope that other researchers will use the tool and actively help us to develop it by sending us their suggestions, or code modifications, for us to incorporate.

5. Methods

5.1. Synthesis of polymer-grafted nanoparticles

Spherical Au NPs were synthesized according to the well-known Turkevich method.[38] The as-synthesized citrate-capped Au nanosphere were then coated with poly(vinyl pyrrolidone) (PVP, M_w = 55k, Sigma-Aldrich) chains in aqueous solution under stirring at room temperature as previously described.[39,40] To remove excess polymer, the Au nanospheres were precipitated by centrifugation (Eppendorf Centrifuge 5804) and redispersed in ethanol. This process was repeated two times. Au nanorods were synthesized by seed-mediated growth as previously reported.[41] The as-synthesized Au nanorods were then coated with PVP polymer in aqueous solution using the same method mentioned above. The ligand exchange process was confirmed by UV-visible absorption spectroscopy, infrared spectroscopy, and dynamic light scattering. PVP-grafted Ag nanocubes were synthesized using a previously reported polyol reaction.[42] The as-synthesized nanocubes were further purified by filtration and concentrated to the desired concentration in pure ethanol as previously described.[3]

5.2. Composite film fabrication

The NP-polymer composite film was fabricated using previously described methods.[3,39] Specifically, about 180–200 nm thick polystyrene (PS, M_w range from 3k to 28k) films were spun coated onto clean Si substrates and film thickness was measured by atomic force microscopy (Veeco, Multimode Nanoscope IV). PVP-coated NPs were typically precipitated in ~100 μL ethanol before diluting with 1–2 ml $CHCl_3$. NP dispersions were then added dropwise to an air-water interface until a floating monolayer of nanoparticles was obtained. The NP monolayer was then transferred onto supported PS thin-films by dip-coating. The NP-PS composite was then exposed to $CHCl_3$ vapor in a closed vessel at room temperature according to previous methods.[33,43] For monitoring the time-dependent evolution of assembly structures, the nanocomposite films were enclosed in individual vessels during the vapour exposure step and removed from the vessel after the desired time interval. Assembly time is limited by degradation of the NP-polymer composite. The metal NPs experience significant polymer dewetting at later assembly times, which stops growth altogether by preventing NP diffusion and pinning the NPs against the underlying solid support (this occurs earlier in experiments E3 and E4, as noted from the growth curves in Fig. 7). This dewetting is primarily controlled by polymer-NP interactions that are difficult to control uniformly across NPs with different sizes, shapes, and compositions.

5.3. Sample characterization

NP assembly was characterized by scanning electron microscopy (SEM) using a FEI UHR Field Emission SEM equipped with a field emission cathode with a lateral resolution of approximately 2 nm. The acceleration voltage was chosen between 5 and 30 kV.

List of symbols

s	Cluster size (# of constituent particles)
t	Experiment time
$\Phi(t)$	Total NP area fraction in an image
w_i	Bias-correction weight for cluster i [eqn (2)]
$N_s(t)$	Cluster size distribution (# of s-particle clusters)
$\nu_s(t)$	Relative cluster size distribution [eqn (3)] (# of clusters/total # of NPs)
$n_s(t)$	Cluster number concentration (# of s-particle clusters/unit area)
$S_1(t)$	Number-average mean cluster size [eqn (4)]
$S_2(t)$	Mass-average mean cluster size [eqn (5)]
$p(a)$	Distribution of NP (projection) areas
$R_1{}^2$, $R_2{}^2$	Principal moments of the gyration tensor
R_g	Radius of gyration [eqn (10)]
AR	Aspect ratio [eqn (11)]
A_1	Anisotropy ratio [eqn (12)]
ε	Elongation (relative anisotropy) [eqn (13)]
κ	Fractional anisotropy [eqn (14)]
L	Backbone length (graph diameter)
L_{ee}	End-to-end distance
$D_f^{(R)}$	Fractal scaling dimension of s with R_g [eqn (16)]
d_L	Scaling dimension of s with L [eqn (17)] ("chemical" or "spreading" dimension)
$D_f^{(C)}$	Fractal dimension obtained from decay of cluster radial cumulative distribution function [eqn (18)]

Acknowledgements

This work was supported through a grant to ART and GA from the National Science Foundation (CMMI, Award no. 1200850). GA also acknowledges the donors of the American Chemical Society Petroleum Research Fund (Award no. 52515-ND7) for partial support of this research. CRM was supported through a Jacobs Graduate Fellowship from UCSD.

References

1 L. Dai and A. W. Mau, *Adv. Mater.*, 2001, **13**, 899–913.

2 I. Gur, N. A. Fromer, C.-P. Chen, A. G. Kanaras and A. P. Alivisatos, *Nano Lett.*, 2007, **7**, 409–414.

3 B. Gao, G. Arya and A. R. Tao, *Nat. Nanotechnol.*, 2012, **7**, 433–437.

4 B. Gao, M. J. Rozin and A. R. Tao, *Nanoscale*, 2013, **5**, 5677–5691.

5 S. R. Forrest and T. A. Witten Jr., *J. Phys. A: Math. Gen.*, 1979, **12**, L109–L117.

6 M. Lapuerta, O. Armas and A. Gómez, *Aerosol Sci. Technol.*, 2003, **37**, 369–381.

7 R. Fisker, J. M. Carstensen, M. F. Hansen, F. Bødker and S. Mørup, *J. Nanopart. Res.*, 2000, **2**, 267–277.

8 P. H. F. Hansen and L. Bergström, *J. Colloid Interface Sci.*, 1999, **218**, 77–87.

9 G. H. Woehrle, J. E. Hutchison, S. Ozkar and R. G. Finke, *Turk. J. Chem.*, 2006, **30**, 1–13.

10 M. D. Abràmoff, P. J. Magalhães and S. J. Ram, *Biophotonics Int.*, 2004, **11**, 36–43.

11 D. Robinson and J. Earnshaw, *Phys. Rev. A*, 1992, **46**, 2045–2053.

12 E. Verleysen, P.-J. De Temmerman, E. Van Doren, M. Abi Daoud Francisco and J. Mast, *Powder Technol.*, 2014, **258**, 180–188.

13 P.-J. De Temmerman, E. Verleysen, J. Lammertyn and J. Mast, *Powder Technol.*, 2014, **261**, 191–200.

14 PICT, Matlab Exchange Server, http://www.mathworks.com/matlabcentral/fileexchange/49382-pict–particle-image-characterization-tool, (accessed Jan 28, 2015).

15 S. Eddins, *Almost-connected-component-labeling*, 2010, http://blogs.mathworks.com/steve/2010/09/07/almost-connected-component-labeling/, (accessed Jan 28, 2015).

16 J. C. Russ, *The Image Processing Handbook*, CRC press, 5th edn, 2010.

17 M. von Smoluchowski, *Z. Phys. Chem*, 1917, **92**, 129–168.

18 T. Vicsek and F. Family, *Phys. Rev. Lett.*, 1984, **52**, 1669–1672.

19 *Colloidal Particles at Liquid Interfaces*, ed. B. P. Binks and T. S. Horozov, Cambridge University Press, 2006.

20 R. Jullien and R. Botet, *Aggregation and Fractal Aggregates*, World Scientific, 1987, p. 130.

21 G. E. P. Box and G. C. Tiao, *Bayesian inference in statistical analysis*, John Wiley & Sons, 2011, vol. 40.

22 R. Botet and R. Jullien, *J. Phys. A: Math. Gen.*, 1986, **19**, L907–L912.

23 R. Mukundan and K. R. Ramakrishnan, *Moment functions in image analysis: theory and applications*, World Scientific, 1998, p. 150.

24 G. C. Bushell, Y. D. Yan, D. Woodfield, J. Raper and R. Amal, *Adv. Colloid Interface Sci.*, 2002, **95**, 1–50.

25 B. B. Mandelbrot, *The fractal geometry of nature*, Macmillan, 1983, vol. 173.

26 R. Jullien, R. Botet and P. M. Mors, *Faraday Discuss. Chem. Soc.*, 1987, **83**, 125–137.

27 H. J. Herrmann, *Phys. Rep.*, 1986, **136**, 153–227.

28 X. Xiao, E. P. White, M. B. Hooten and S. L. Durham, *Ecology*, 2011, **92**, 1887–1894.

29 B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*, CRC Press, 1994, vol. 57.

30 D. F. Gatz and L. Smith, *Atmos. Environ.*, 1995, **29**, 1185–1193.

31 R. C. Ball, D. A. Weitz, T. A. Witten and F. Leyvraz, *Phys. Rev. Lett.*, 1987, **58**, 274–277.

32 K. Liu, Z. Nie, N. Zhao, W. Li, M. Rubinstein and E. Kumacheva, *Science*, 2010, **329**, 197–200.

33 P. Akcora, H. Liu, S. K. Kumar, J. Moll, Y. Li, B. C. Benicewicz, L. S. Schadler, D. Acehan, A. Z. Panagiotopoulos, V. Pryamitsyn, *et al.*, *Nat. Mater.*, 2009, **8**, 354–359.

34 H. J. Herrmann and H. E. Stanley, *J. Phys. A: Math. Gen.*, 1988, **21**, 829–833.

35 T. A. Witten and L. M. Sander, *Phys. Rev. Lett.*, 1981, **47**, 1400–1403.

36 P. Meakin, *Phys. Rev. Lett.*, 1983, **51**, 1119–1122.

37 M. Kolb, R. Botet and R. Jullien, *Phys. Rev. Lett.*, 1983, **51**, 1123–1126.

38 J. Turkevich, P. C. Stevenson and J. Hillier, *Discuss. Faraday Soc.*, 1951, **11**, 55–75.

39 B. Gao, Y. Alvi, D. Rosen, M. Lav and A. R. Tao, *Chem. Commun.*, 2013, **49**, 4382–4384.

40 M. A. Correa-Duarte, J. Pérez-Juste, A. Sánchez-Iglesias, M. Giersig and L. M. Liz-Marzán, *Angew. Chem., Int. Ed.*, 2005, **44**, 4375–4378.

41 N. R. Jana, L. Gearheart and C. J. Murphy, *Adv. Mater.*, 2001, **13**, 1389.

42 A. Tao, P. Sinsermsuksakul and P. Yang, *Angew. Chem., Int. Ed.*, 2006, **45**, 4597–4601.

43 J. B. Hooper, D. Bedrov and G. D. Smith, *Langmuir*, 2008, **24**, 4550–4557.