IIE Transactions

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/uiie20

A multistage, semi-automated procedure for analyzing the morphology of nanoparticles

Chiwoo Park [a], Jianhua Z. Huang [a], David Huitink [a], Subrata Kundu [a], Bani K. Mallick [a], Hong Liang [a] & Yu Ding [a]

[a] Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX, 77843, USA

Available online: 21 Jun 2011

To cite this article: Chiwoo Park, Jianhua Z. Huang, David Huitink, Subrata Kundu, Bani K. Mallick, Hong Liang & Yu Ding (2012): A multistage, semi-automated procedure for analyzing the morphology of nanoparticles, IIE Transactions, 44:7, 507-522

To link to this article: http://dx.doi.org/10.1080/0740817X.2011.587867

IIE Transactions (2012) **44**, 507–522
Copyright © "IIE"
ISSN: 0740-817X print / 1545-8830 online
DOI: 10.1080/0740817X.2011.587867

A multistage, semi-automated procedure for analyzing the morphology of nanoparticles

CHIWOO PARK, JIANHUA Z. HUANG, DAVID HUITINK, SUBRATA KUNDU, BANI K. MALLICK, HONG LIANG and YU DING*

Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX 77843, USA
E-mail: yuding@iemail.tamu.edu

Received October 2010 and accepted April 2011

This article presents a multistage, semi-automated procedure that can expedite the morphology analysis of nanoparticles. Material scientists have long conjectured that the morphology of nanoparticles has a profound impact on the properties of the hosting material, but a bottleneck is the lack of a reliable and automated morphology analysis of the particles based on their image measurements. This article attempts to fill in this critical void. One particular challenge in nanomorphology analysis is how to analyze the overlapped nanoparticles, a problem not well addressed by the existing methods but effectively tackled by the method proposed in this article. This method entails multiple stages of operations, executed sequentially, and is considered semi-automated due to the inclusion of a semi-supervised clustering step. The proposed method is applied to several images of nanoparticles, producing the needed statistical characterization of their morphology.

Keywords: Nano imaging, morphology analysis, shape analysis, machine learning, nanoparticle overlapping

1. Introduction

Material scientists have conjectured that the morphology of nanoparticles has a profound impact on the properties of the hosting material; see, for example, Wang *et al.* (1998), Mohamed *et al.* (2000), El-Sayed (2001), Nehl *et al.* (2006), and Pan *et al.* (2007). After a synthesis process of nanoparticles, measurements can be taken using some nano-specializing metrology devices (such as electron microscopes). The outputs from the metrology devices are gray-scale images of nanoparticles and its surrounding material in a sampled region; please see Fig. 1 for examples. These images need to be processed to yield meaningful morphological parameters, characterizing the shape and size of the nanoparticles; this is known as *morphology analysis*.

A bottleneck in such research endeavors is the lack of a reliable, efficient, and automated process for the characterization and quantification of the size and shape of the nanoparticles, based on the nanoparticle images (hereafter shortened to "nano images"). Through our communication with several research groups in nanotechnology in and outside the United States, we understand that the current practice of morphology analysis is still largely a manual counting process, aided by certain software tools that are not specifically designed for handling nano images. The popular tools include `ImageJ` (http://rsbweb.nih.gov/ij) and `AxioVision` (http://www.zeiss.com/). `ImageJ` is popular probably because it is a freeware tool provided by the National Institutes of Health for cell morphology analysis. There are certain similarities between bio images and nano images. It is not surprising that people went to the bioimaging field to look for a tool. However, when `ImageJ` is applied to the nano images in Fig. 1, the particle recognition rates are about 28% (left) and 48% (right), respectively, deemed by the domain experts in our research team as too low for the purpose of generating statistically reliable and representative morphological results. The results from `ImageJ` are presented in Section 7.

Despite the importance of morphology measurements, there is only a limited amount of literature about automated morphology analysis of nanomaterials. All of them used circularity of particle's contours (McFarland and Van Duyne, 2003; Glotov, 2008; Chen, 2009) or an elliptical shape template (Fisker *et al.*, 2000) to segment overlapping particles, so their applications are limited.

There is a rich body of literature dealing with similar problems, especially in the field of biomedical imaging. However, the majority of the literature in biomedical imaging concentrates on locating cells in micrographs (Sage *et al.*, 2005; Jiang *et al.*, 2007) or separating uniformly shaped cells, usually elliptical cells, from the background (Jung *et al.*, 2008; Kothari *et al.*, 2009). Applying existing methods from bio-imaging to the recognition of

*Corresponding author

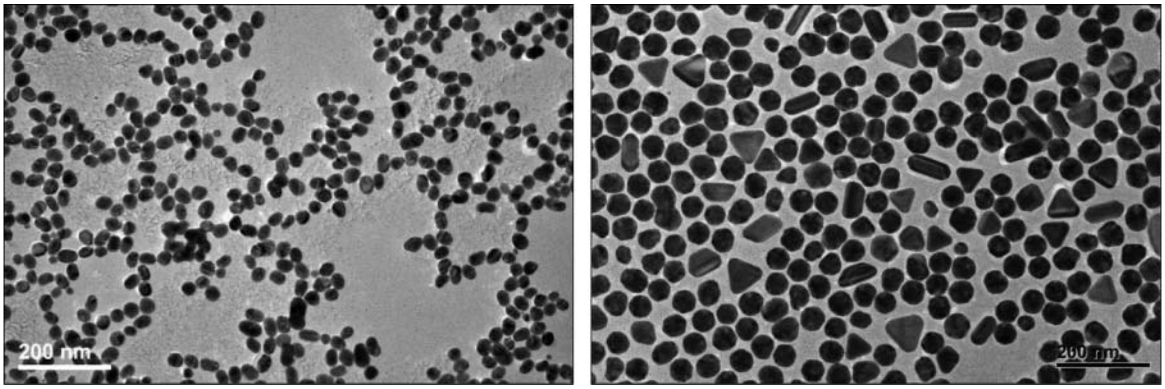

Fig. 1. Example transmission electron microscope images.

nanoparticles is ineffective primarily because of a phenomenon called *particle touching*, namely, that the nanoparticles often overlap with one another to varying degrees. By contrast, cell morphology analysis in bioimaging frequently works on a single cell because it usually pays to manually isolate a cell from its surrounding tissues. We could find papers dealing with several cells in one image but usually falling into the situations that either the cells are well separated, or the cells, though overlapped, are shaped elliptically (Jung *et al.*, 2008). Once the shape is fixed to elliptical, it is relatively easy to separate cells by testing the roundness of the objects detected.

Our objective is to devise an effective procedure that can attain a much higher recognition rate of nanoparticles in nano images. By accomplishing high recognition, we are able to obtain statistically more reliable distributions for sizes and shapes of nanoparticles. Our basic strategy for attaining a high recognition is to first learn and construct shape statistics from those clearly identifiable particles and then use the shape statistics to perform statistical reasoning on those nanoparticles insufficiently informed by nano images.

The shape statistics implies the variations in shapes within each predefined shape categories. The procedure to construct the shape statistics is described in the bottom part of Fig. 2. It starts off by extracting the boundaries of nanoparticles, which contain sufficient information for their morphology. If a particle is well separable from the background and thus its boundary can be completely extracted, the procedure is to extract the shape features from the boundary and to determine the shape class to which the feature belongs. There are two major challenges. The first one is to extract *low-dimensional* shape features invariant to undesired variations; e.g., rotation, shift, and scaling of shapes. We propose a non-linear projection to map boundaries to shape features having the invariance property (Sections 3 and 4). The second challenge is to classify a given shape feature into a set of predetermined

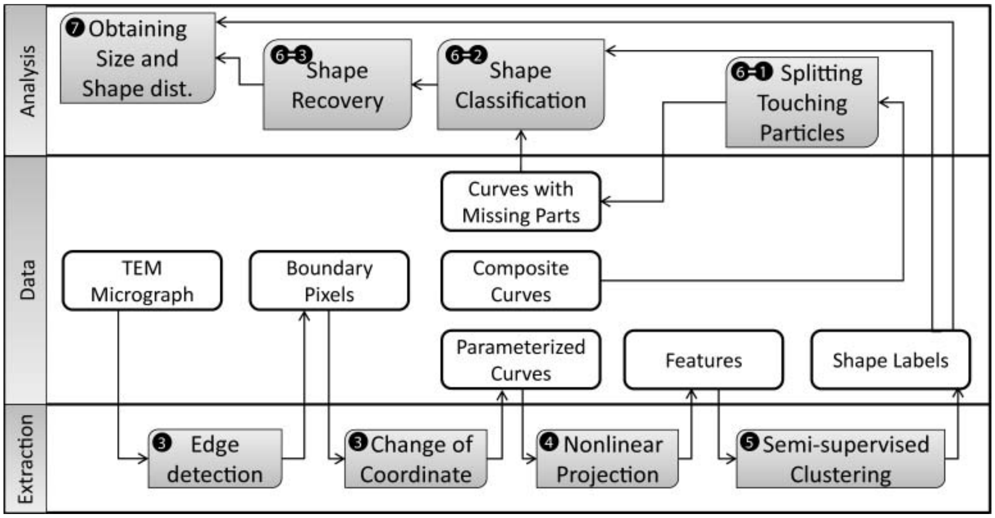

Fig. 2. Logical flow of the morphology analysis: white boxes depict the different representations of morphology data. Gray boxes represent the steps in the procedure for data representation and analysis. The number in each gray box refers to a section in this article. Arrows show the logical input–output between the white and gray boxes.

shape classes with minimal human intervention. For that, we use a multi-class semi-supervised clustering method (Section 5).

If a particle overlaps with other particles and its boundary is barely separable from the background, we need more analysis on the particle (Section 6). This part is reflected in the top part of Fig. 2. For the case of overlaps, the boundaries from many particles compose a complicated pattern of composite boundaries (see Fig. 7). We develop a boundary-split method that separates the composite boundaries into several simple boundaries, one for each particle. Some parts of the separated boundary are missing because of occlusion by other particles. We then perform a statistical reasoning to recover the missing parts by using a procedure based on the Functional Principal Component Analysis (FPCA).

This article provides a multistage procedure for extracting morphological information on nanoparticles. Because of the complicated nature of the real application, it is necessary for our procedure to consist of multiple steps. Our contribution is to sort out the necessary steps and to identify the statistical methods to apply in order to solve the practical problem. The purpose of this article is not the development of brand new statistical methods but rather the novel application of existing ones in this emerging engineering problem of nano-imaging. Our procedure incorporates several statistical learning tools including multidimensional scaling, semi-supervised clustering, multi-class classification, peak detection, and FPCA. While each statistical method focuses on one aspect of the problem, what provides the complete solution is the appropriate integration of all the components.

The rest of this article is organized as follows. Section 2 provides more details regarding nano-imaging. Sections 3 to 5 explain an affine-invariant shape feature space along with the semi-supervised shape clustering method working in the feature space. Section 6 is about the statistical inference on particle boundaries partially hidden in nano images. Finally, Section 7 presents the size and shape distributions obtained by our method as well as comparisons with those from `ImageJ`. Section 8 concludes this article.

2. Transmission electron microscopy

The particular nano metrology device used in this research for analyzing the nanoparticles is the Transmission Electron Microscope (TEM). The nanoparticles we analyzed were mostly gold particles in a water-based solution. In order to observe the morphology of nanoparticles, a drop of the solution was deposited on a sample holder; i.e., a TEM carbon grit. After the water had evaporated, the nanoparticles were observed using the TEM. A JEOL 2010 high-resolution TEM operating at 200 kV accelerating voltage was used, which has a 0.27-nm point resolution. The microscope transmits a beam of electrons through the particle-deposited grit such that a gray-scale image was obtained. Usually, one pixel of the gray-scale image has 256 possible gray-scale values. Refer to Fig. 1 for examples of TEM images.

Due to the absorption of electrons by atoms, the regions occupied by the nanoparticles usually look darker in the image. The darkness pattern may be related to the crystal structure and/or thickness of nanoparticles. Additionally, one can see many tiny dark dots in the background, which are uniformly distributed throughout the image region. These dark dots are generated because the atoms of the carbon grit also absorb electrons. One may also notice a thin white area wrapping around the whole or partial boundary of a particle. This is the result of having surfactants on the rim of the particle. The surfactants are added to alleviate the aggregating effect among particles in the process of synthesis.

3. Representation of particle boundaries

To extract the size and shape distributions of nanoparticles in a TEM image, we need first to recognize the boundaries of particles and have a convenient mathematical representation of these boundaries. We address two issues in this section: (a) extracting the particle boundaries in a TEM image and (b) representing the boundaries using parametric curves.

To obtain particle boundaries, we use an established edge detection technique. Edge detection is a research topic that has been thoroughly studied in the image processing literature. We chose to use Canny's algorithm (Canny, 1986), one of the mature algorithms having a high sensitivity in edge detection. The edges detected by applying Canny's algorithm to a TEM image are in a far greater number than needed for forming the boundaries of nanoparticles. We apply a simple thresholding rule (Gonzalez and Woods, 2002, pp. 760–769) that can remove the unnecessary edges.

Every extracted boundary is in the form of a set of pixel locations. To make the subsequent shape analysis easy, we change the boundary representation into a parametric curve. For shape analysis, a basic requirement for the parametric curve is invariance under rotation, translation, and scaling; that is, the set of parameters representing a shape does not change when an object of the same shape rotates, translates, or changes size.

To meet the invariance requirement, we parameterize a particle boundary by modifying the polar coordinate system. In this parameterization, a boundary is represented by a set of pairs (r_i, θ_i), where r_i is the normalized distance of the ith point on the boundary to the gravity center of the particle, and θ_i is the angle between a prespecified axis and the directed line connecting the gravity center and the ith point. If R_i is the physical distance from the gravity center to the ith point, the normalized distance r_i is defined as $r_i = R_i/\mu_R$, where μ_R is the sample average of R_i. Note that using the sample average of R_i is a popular way to

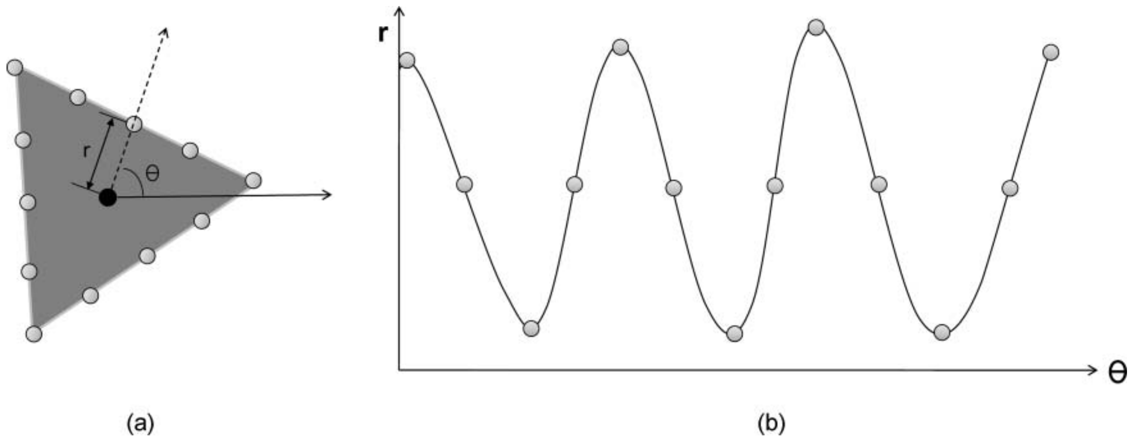

Fig. 3. Representation of nanoparticle boundaries; in (a), the black dot is the gravity center of the triangle and the gray points are pixels sampled from the boundary of the triangle, which correspond to the gray points in (b).

measure the size of a shape (called *centroid size*) for the purpose of scaling (Dryden and Mardia, 1998, pp. 23–24).

As shown in Fig. 3, the result of this parameterization is a curve (or a set of functional data) in the r–θ coordinate system. Since both r and θ are defined relative to the gravity center, they are invariant to translation. This parameterization is scale invariant because the distance is normalized and the angle is not influenced by scaling. However, the parameterization is not yet rotation invariant; this issue will be addressed later in Section 4.

This parameterization characterizes convex shapes such as polygons and circles very well. For example, the curve with three modes shown in Fig. 3(b) corresponds to the triangular shape in Fig. 3(a). Similarly, the representing curve has four modes for a rectangle and so forth for other polygons. Note that, if we use the same set of angles to represent all particles and the angles are taken to be evenly spaced over $[0, 2\pi]$, we only need to record the values of the distances r_i.

For non-convex shapes, this parameterization is effective only for star shapes. Fortunately, the shapes of nanoparticles in our applications are mostly convex. There is a physical explanation behind this phenomenon. A high surface-to-volume ratio provides a strong driving force to speed up the thermodynamic processes that minimize thermodynamic free energy and, as a result, materials with a high surface-to-volume ratio are not stable. Since convex shapes have smaller surface-to-volume ratios than non-convex shapes, the shapes of nanoparticles are prone to being stabilized to convex shapes. For this reason, the use of our parameterization is appropriate for the analysis of nanoparticles.

4. Feature extraction by non-linear dimension reduction

To analyze the variations of shape, we need to obtain a sufficient number of parametric curves in each shape class, which demands a shape clustering method. There are two technical difficulties hindering accurate shape clustering. First, many clustering methods rely on using a distance or similarity measure between the objects to be clustered; however, for morphology analysis there is no straightforward definition of the similarity measure between a pair of parametric curves because a parametric curve is rotationally variant. Second, the dimension of the resulting parametric curves from a particle is high. It is well known that clustering analysis methods using a similarity measure work poorly in high-dimension spaces (Steinbach *et al.*, 2003, pp. 12–13).

In this section, we provide a solution that addresses these two difficulties. We define a rotationally invariant similarity measure on the space of parametric curves and a non-linear projection of the parametric curves to a low-dimensional Euclidean space using the technique of Isomap (Tenenbaum *et al.*, 2000; Choi and Choi, 2007). These two components are combined in one procedure, to be explained below, so that the end result of this procedure is a dimension reduced, rotationally invariant feature set.

Given m parametric curves $\mathbf{f}_1, \ldots, \mathbf{f}_m$, each represents a boundary of a particle. If each curve is evenly sampled at every $2\pi/n$ for the angle parameter θ, then the curve can be represented by a vector; e.g., $\mathbf{f}_i = (r_{i1}, r_{i2}; \ldots, r_{in})^{\mathrm{t}}$. A large value of n will ensure the accuracy of this vector representation of the curves but will create problems for subsequent clustering task. We want to project $\mathbf{f}_i$ onto a low-dimensional space by an embedding map $\boldsymbol{\phi} : \mathcal{R}^n \rightarrow \mathcal{R}^p$ such that $p \ll n$.

Recall that we want $\boldsymbol{\phi}(\mathbf{f}_i)$ to be rotationally invariant. Rotating the particle i by the angle of $2\pi/n$ clockwise corresponds to shifting the elements of $\mathbf{f}_i$ circularly downward by one. Thus, the requirement of rotational invariance on $\boldsymbol{\phi}(\mathbf{f}_i)$ can be expressed as

$$\boldsymbol{\phi}(\mathbf{f}_i) = \boldsymbol{\phi}(s_t \circ \mathbf{f}_i) \qquad \text{for all } t = 1, 2, \ldots, n, \tag{1}$$

where $s_t \circ \mathbf{v}$ is an operator circularly shifting each element of $\mathbf{v}$ downward by t elements, which is defined by $s_t \circ \mathbf{v} := (v_{t+1}, \ldots, v_n, v_1, \ldots, v_t)^{\mathrm{t}}$ for $\mathbf{v} = (v_1, \ldots, v_n)^{\mathrm{t}} \in \mathbb{R}^n$.

Finding a mapping $\boldsymbol{\phi}$ meeting the constraints in Equation (1) is generally difficult. It is considerably easier, however, to define a rotationally invariant distance between a pair of curves $\mathbf{f}_i$ and $\mathbf{f}_j$. For this reason, toward the objective of finding a dimension reduced, rotationally invariant mapping $\boldsymbol{\phi}$, our strategy is to first define a rotationally invariant distance, then create a dissimilarity matrix using such distance, and finally apply the MultiDimensional Scaling (MDS) technique (Kruskal and Wish, 1978) to obtain this $\boldsymbol{\phi}$.

Given two curves $\mathbf{f}_i$ and $\mathbf{f}_j$, define a rotationally invariant distance as

$$d_{ij} := d(\mathbf{f}_i, \mathbf{f}_j) = \min_t \|\mathbf{f}_i - s_t \circ \mathbf{f}_j\|, \tag{2}$$

where $\|\cdot\|$ is the Euclidean distance. Since the collection of particle shapes may form a curved manifold structure (see Fig. 5), we use the geodesic distance rather than the Euclidean distance to define the dissimilarity matrix before applying the MDS technique (Tenenbaum *et al.*, 2000). The geodesic distance is the distance between two points over the curvature in a manifold and reflects the non-linear structure of the data distribution. For neighboring points, the Euclidean distance provides a good approximation to the geodesic distance, while for far apart points, the geodesic distance can be approximated by adding a sequence of local hops between neighboring points. As such, to compute the geodesic distances, we first construct a graph of data points having edges with non-zero weights d_{ij} for the k-nearest neighbors only. Then define the geodesic distance as the distance of the shortest path between a pair of two data points. In the following, $g(\cdot,\cdot)$ is used to denote the geodesic distance.

To illustrate, see the left subfigure in Fig. 4. Suppose that all the $\mathbf{f}_i$ are in a p-dimensional subspace embedded in $\mathcal{R}^n$ (also called p-manifold.) The subspace is approximated by a surface generated from the meshes of the k-nearest neighborhood graph, so that the distance in the subspace is the distance in the graph; this distance is the geodesic distance $g(\cdot,\cdot)$. In the figure, the geodesic distance between $\mathbf{f}_i$ and $\mathbf{f}_j$ is the summation of the weights of the edges that are on the shortest path from $\mathbf{f}_i$ to $\mathbf{f}_j$ on the graph.

Given the dissimilarities $g_{ij} = g(\mathbf{f}_i, \mathbf{f}_j)$ produced by the geodesic distance, we find the low-dimensional features $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_m$ of curves $\mathbf{f}_1, \ldots, \mathbf{f}_m$ such that the Euclidean distances between the features are close to the corresponding geodesic distances between the curves; that is,

$$g_{ij}^2 \approx (\boldsymbol{\phi}_i - \boldsymbol{\phi}_j)^{\mathrm{t}}(\boldsymbol{\phi}_i - \boldsymbol{\phi}_j). \tag{3}$$

Denote by $\mathbf{G}$ the dissimilarity matrix whose (i, j)-entry is g_{ij} and denote the doubly centered geodesic distance matrix.

$$\tau(\mathbf{G}^2) = -\frac{1}{2}\mathbf{H}\mathbf{G}^2\mathbf{H}^{\mathrm{t}}, \tag{4}$$

where $\mathbf{G}^2$ is the matrix whose elements are the squares of the elements of $\mathbf{G}$, and $\mathbf{H}$ is the $m \times m$ centering matrix with $(\mathbf{H})_{ij} = \delta_{ij} - (1 - m)$. If the kernel matrix $\tau(\mathbf{G}^2)$ is positive semi-definite, the classical MDS technique gives an explicit solution to the embedding problem. Let $\tau(\mathbf{G}^2) = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{t}}$ be the eigen-decomposition of $\tau(\mathbf{G}^2)$, then the collection of p-dimensional features $\mathbf{X} = [\phi_1, \ldots, \phi_m]^{\mathrm{t}}$ is given by $\mathbf{X} = \mathbf{V}[\,,1:p]\,\boldsymbol{\Lambda}[1:p, 1:p]^{1/2}$, where $\mathbf{V}[\,,1:p]$ is the first p columns of $\mathbf{V}$ and $\boldsymbol{\Lambda}[1:p, 1:p]$ is the $p \times p$ upper-left corner of $\boldsymbol{\Lambda}$. Unfortunately, because of the use of geodesic distances in defining $\mathbf{G}$, the matrix $\tau(\mathbf{G}^2)$ is not guaranteed to be positive semi-definite. As a remedy, we use the constant-shifting method that is well studied in the metric MDS and replace $\mathbf{G}$ in Equation (4) by the matrix $\widetilde{\mathbf{G}}$ with entries $\tilde{g}_{ij} = g_{ij} + c(1 - \delta_{ij})$, where δ_{ij} is the Kronecker delta, and c is the largest eigenvalue of the matrix:

$$\begin{bmatrix} 0 & 2\tau(\mathbf{G}^2) \\ -\mathbf{I} & -4\tau(\mathbf{G}) \end{bmatrix}. \tag{5}$$

According to Cailliez (1983), the matrix $\tau(\widetilde{\mathbf{G}}^2)$ is positive semi-definite. Therefore, we can apply the classical MDS

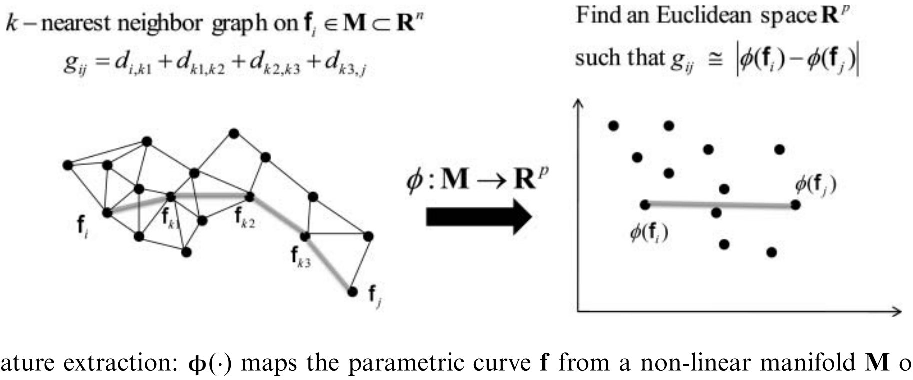
k − nearest neighbor graph on $\mathbf{f}_i \in \mathbf{M} \subset \mathbf{R}^n$
$g_{ij} = d_{i,k1} + d_{k1,k2} + d_{k2,k3} + d_{k3,j}$
Find an Euclidean space $\mathbf{R}^p$
such that $g_{ij} \cong |\phi(\mathbf{f}_i) - \phi(\mathbf{f}_j)|$
$\phi: \mathbf{M} \to \mathbf{R}^p$
$\mathbf{f}_i$
$\mathbf{f}_{k1}$
$\mathbf{f}_{k2}$
$\mathbf{f}_{k3}$
$\mathbf{f}_j$
$\phi(\mathbf{f}_j)$
$\phi(\mathbf{f}_i)$

Fig. 4. Basic idea of feature extraction: $\boldsymbol{\phi}(\cdot)$ maps the parametric curve $\mathbf{f}$ from a non-linear manifold $\mathbf{M}$ onto a low-dimensional Euclidean space $\mathcal{R}^p$, such that the Euclidean distance between the transformed curves $\boldsymbol{\phi}(\mathbf{f}_i)$ and $\boldsymbol{\phi}(\mathbf{f}_j)$ is “close” to the geodesic distance $g(\mathbf{f}_i, \mathbf{f}_j)$ defined on the original manifold $\mathbf{M}$. The p ($\ll n$) elements in $\boldsymbol{\phi}(\mathbf{f})$ are called the p features of the original parametric curve $\mathbf{f}$.

Algorithm 1. Feature Extraction by Isomap.

```
1. Construct a $k$-nearest neighbor graph on
   $\{\mathbf{f}_i; i = 1, \ldots, m\}$.
2. Compute $\mathbf{G}$ with $(\mathbf{G})_{ij} = g(\mathbf{f}_i, \mathbf{f}_j)$.
3. Compute $\tau(\mathbf{G})$ using Equation (4) and compute $c$ by
   taking the largest eigenvalue of matrix (5).
4. Compute $\tau(\widetilde{\mathbf{G}}^2)$ using Equation (4) and also
   substituting in $\widetilde{\mathbf{G}}^2 = \mathbf{G} + c(\mathbf{1}_m - \mathbf{I}_m)$, where $\mathbf{1}_m$ is an
   $m \times m$ matrix of ones and $\mathbf{I}_m$ is an $m \times m$ identity
   matrix.
5. Perform eigen-decomposition: $\tau(\widetilde{\mathbf{G}}^2) = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^t$.
6. Obtain $\mathbf{X} = \mathbf{V}[\ , 1 : p]\, \mathbf{\Lambda}[1 : p, 1 : p]^{1/2}$ for a specific
   dimension $p$.
```

procedure previously described to the modified kernel matrix $\tau(\widetilde{\mathbf{G}}^2)$ to obtain the desired non-linear embeddings. All necessary steps for the non-linear embedding are summarized in Algorithm 1.

In the non-linear embedding method, there are two tuning parameters, k (number of nearest neighbors) and p (the dimension of the reduced space or the embedding), whose choice are important. To compute the geodesic distances, we need decide on k, the number of nearest neighbors. If k is too large, it will cause the "short circuit" edges that shortcut the true geometry of a manifold reflecting the non-linear structure of data; if k is too small, it will cause the manifold to fragment into a large number of disconnected clusters. Following Samko *et al.* (2006), we choose k by maximizing $|\rho(\mathbf{D}, \boldsymbol{\phi}_{k,p})|$, where $\mathbf{D}$ and $\boldsymbol{\phi}_{k,p}$ are the matrices of the Euclidean distances between a pair of points in the original space and the feature space, respectively, and $\rho(\cdot, \cdot)$ is the linear correlation coefficient. Note that $\boldsymbol{\phi}_{k,p}$ depends on p. Samko *et al.* (2006) argued that the dataset has its intrinsic dimension, and subsequently they showed empirically that p does not change even if k changes. Hence, we decide to first estimate p for an arbitrary (but reasonable) choice of k and then choose the optimal k with this p.

In the application to the TEM image in the right-hand panel in Fig. 1, we extracted 420 (i.e., $m = 420$) parameterized curves, where each curve was represented by a 315-dimensional vector (i.e., $n = 315$). Subsequently, we computed the rotationally invariant pairwise distances d_{ij} and constructed the graph structure retaining edges among the k-nearest neighbors with $k = 12$. Finally, we projected $\mathbf{f}$s to a rotationally invariant subspace of dimension three (i.e., $p = 3$) to obtain the features $\boldsymbol{\phi}(\mathbf{f})$. The number of neighbors was chosen by maximizing the correlation criterion given in the previous paragraph. The dimension of the low-dimensional subspace was chosen by using the scree plot of the kernel matrix $\tau(\widetilde{\mathbf{G}}^2)$; see the right-hand panel in Fig. 5. The left-hand panel in Fig. 5 shows the scatterplot of the 420 curves distributed in the feature space.

5. Semi-supervised clustering of shapes

The low-dimensional features obtained by the non-linear embedding can be used as inputs to clustering algorithms to cluster the nanoparticles. However, our experience is that generic clustering methods do not work well in our context. Such methods tend to overly partition a dataset into far more groups than what is needed in nanomaterial research. We thus adopt a semi-supervised learning approach

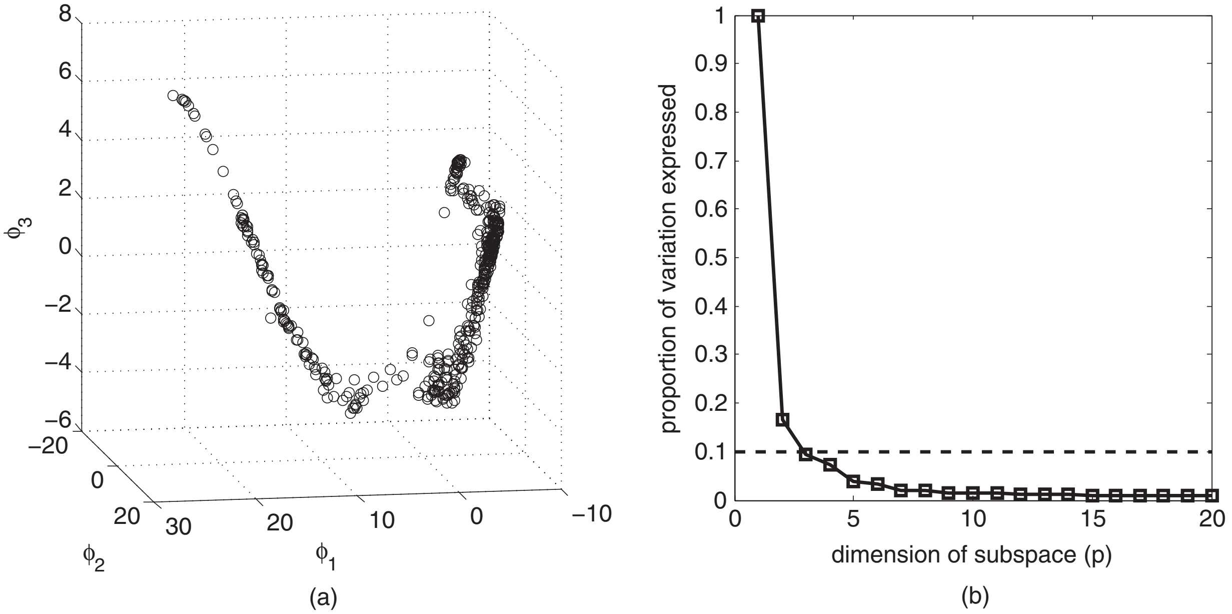

Fig. 5. Rotational invariance feature extraction. The left panel is the scatterplot showing the projection of the 420 curves in the three-dimensional feature space. The right panel shows a scree plot of the kernel matrix.

in which we ask domain experts to determine the number of shape groups and manually pick a small number of particles from each shape group. We then use these labeled cases as training data. It is known that semi-supervised learning is able to significantly increase the accuracy of clustering by using a small number of labeled data, together with a large number of unlabeled data (Zhu, 2005).

There are various semi-supervised learning methods reported in the literature, including using a generative model with the EM algorithm, self-training, information regularization, and graph-based semi-supervised learning; see Zhu (2005) for a comprehensive review. We find it convenient to use the graph-based approach proposed by Zhu *et al.* (2003). The basic strategy is as follows. Labeled and unlabeled data are represented as vertices in a connected graph, where each edge is assigned a weight that measures the similarity between the two data points connected by the edge; the method produces a label function that is smooth on the graph and correctly matches the known label. However, the method was originally designed for binary classification; here we extend it for multi-class classification.

Suppose that we have l labeled points $(\boldsymbol{\phi}_1, t_1), \ldots, (\boldsymbol{\phi}_l, t_l)$ from K classes and u unlabeled points $(\boldsymbol{\phi}_{l+1}, t_{l+1}), \ldots, (\boldsymbol{\phi}_m, t_m)$ with $m = l + u$, where $\boldsymbol{\phi}_i \in \mathcal{R}^p$ is the feature for the ith case and $t_i \in \{0, 1, \ldots, K-1\}$ is the label associated with $\boldsymbol{\phi}_i$; for unlabeled cases, the t_i values are unknown. Construct a connected graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is the collection of nodes corresponding to the m data points, where the coordinates of a data point are specified by $\boldsymbol{\phi}_i$, and $\mathbf{E}$ is the collection of edges. The edge connecting $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$ is weighted by the similarity measure

$$w_{ij} = \exp\left\{-\sum_{d=1}^{p} \frac{(\phi_{id} - \phi_{jd})^2}{\sigma_d^2}\right\},$$

where ϕ_{id} is the dth component of the feature vector $\boldsymbol{\phi}_i$, and σ_d, whose choice will be discussed later, is a scale for the dth feature. Note that the σ_d-scaled Euclidean distance in the feature space corresponds to the geodesic distance in the original (curved) data space by the definition of the feature mapping. Therefore, the weightings, w_{ij}, closely reflect the similarities between parametric curves in the original data space. Our defining of w_{ij} in the feature space is a major difference from Zhu *et al.* (2003), in which the weightings are defined in the original data space.

We construct a vector-valued label function $\mathbf{h} = (h_0, \ldots, h_{K-1})$ on $\mathbf{G}$, taking the feature vector $\boldsymbol{\phi}$ as its input. We require that the label function reproduce the true label for the labeled data; that is, $h_k(\boldsymbol{\phi}_i) = \delta_{t_i,k}$, $i = 1, \ldots, l$, where δ is the Kronecker delta. For unlabeled data, it is assigned the label k^* if $h_{k^*}(\boldsymbol{\phi}) = \max_{k=0,\ldots,K-1} h_k(\boldsymbol{\phi})$. Moreover, it is desirable that we choose the label function $\mathbf{h}$ such that unlabeled points have the same labels as their neighboring points in the graph. These considerations motivate us to obtain the label function by minimizing with respect to $\mathbf{h}$ the following loss function:

$$\begin{aligned} L(\mathbf{h}) &= \frac{1}{2}\sum_{i,j} w_{ij} \|\mathbf{h}(\boldsymbol{\phi}_i) - \mathbf{h}(\boldsymbol{\phi}_j)\|^2 \\ &= \frac{1}{2}\sum_{k}\sum_{i,j} w_{ij} \{h_k(\boldsymbol{\phi}_i) - h_k(\boldsymbol{\phi}_j)\}^2, \end{aligned}$$

subject to the constraints that $h_k(\boldsymbol{\phi}_i) = \delta_{t_i,k}$, $i = 1, \ldots, l$.

Let $\mathbf{W}$ denote the matrix whose (i, j)th entry is w_{ij} and let $\mathbf{h}_k = (h_k(\boldsymbol{\phi}_1), \ldots, h_k(\boldsymbol{\phi}_m))^{\mathrm{t}}$. The loss function can be rewritten as a quadratic form:

$$L(\mathbf{h}) = \frac{1}{2}\sum_{k} \mathbf{h}_k^{\mathrm{t}}\, \Delta\, \mathbf{h}_k, \tag{6}$$

where $\boldsymbol{\Delta} = \mathbf{D} - \mathbf{W}$, and $\mathbf{D}$ is the $m \times m$ diagonal matrix whose ith diagonal entry is $d_i = \sum_j w_{ij}$. To present the solution of this minimization problem, we write the matrices $\mathbf{W}$ and $\mathbf{D}$ and the vector $\mathbf{h}_k$ in block forms, corresponding to labeled parts and unlabeled parts, respectively:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{ll} & \mathbf{W}_{lu} \\ \mathbf{W}_{ul} & \mathbf{W}_{uu} \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \mathbf{D}_{ll} & \mathbf{O} \\ \mathbf{O} & \mathbf{D}_{uu} \end{bmatrix}, \quad \mathbf{h}_k = \begin{bmatrix} \mathbf{h}_k^{(l)} \\ \mathbf{h}_k^{(u)} \end{bmatrix}, \tag{7}$$

where the $\mathbf{O}$s denote matrices of zeros whose dimensions can be determined from the context. Note that $\mathbf{h}_k^{(l)}$ is given by the constraint $h_k(\boldsymbol{\phi}_i) = \delta_{t_i,k}$ for $i = 1, \ldots, l$. Ignoring the term that is completely determined by $\mathbf{h}_k^{(l)}$, the loss function can be written as

$$\frac{1}{2}\big(\mathbf{h}_k^{(u)}\big)^{\mathrm{t}}(\mathbf{D}_{uu} - \mathbf{W}_{uu})\mathbf{h}_k^{(u)} - \big(\mathbf{h}_k^{(l)}\big)^{\mathrm{t}}\mathbf{W}_{ul}\mathbf{h}_k^{(u)}.$$

We get the following closed-from expression for the minimized $\mathbf{h}_k^{(u)}$:

$$\mathbf{h}_k^{(u)} = (\mathbf{D}_{uu} - \mathbf{W}_{uu})^{-1}\mathbf{W}_{ul}^{\mathrm{t}}\mathbf{h}_k^{(l)}. \tag{8}$$

Given the label function in Equation (8), we assign the label $k^* = \arg\max_k h_k(\boldsymbol{\phi}_{l+i})$ to the unlabeled feature $\boldsymbol{\phi}_{l+i}$ for $i = 1, \ldots, u$.

To choose a suitable scale σ_d in the weighting function, we extend the heuristic rule by Zhu *et al.* (2003) to the multi-class setting. We prefer a σ_d that can make the most confident decision of labels. For a K-vector $\mathbf{p}$ with $\mathbf{p}'\mathbf{1}_K = 1$, the Shannon's entropy $H[\mathbf{p}]$ measures the uncertainty associated with the random variable whose probability distribution is $\mathbf{p}$. Thus, a small value of $H[\mathbf{p}]$ is associated with a case that the random variable is more focused on certain value. Since the label function after appropriate normalization induces a probability distribution on the labels, we propose to find σ_d by minimizing the following average entropy:

$$H(\mathbf{h}) := \frac{1}{u}\sum_{i=l+1}^{l+u} H_i\{\mathbf{h}(\boldsymbol{\phi}_i)\}, \tag{9}$$

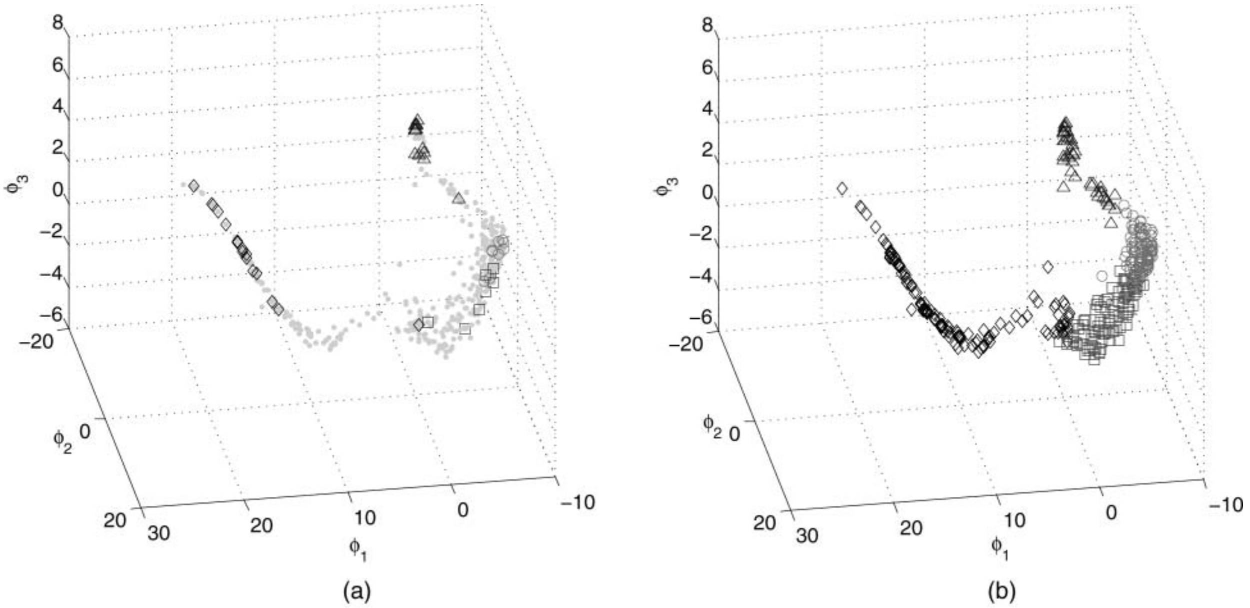

Fig. 6. Results of the semi-supervised learning: Each marker represents a low-dimensional embedding of a parametric curve. The triangular, rectangular, circular, and diamond-shaped markers represent respectively triangular, rectangular, circular, and rod-shaped particles. (a) The initial assign of labels shows a mix of a few manually assigned shape labels and the unassigned ones marked using the small dots and (b) the situation after the semi-supervised learing process.

where

$$H_i\{\mathbf{h}(\boldsymbol{\phi}_i)\} = H\left[\frac{h_0(\boldsymbol{\phi}_i)}{\sum_k h_k(\boldsymbol{\phi}_i)}, \frac{h_1(\boldsymbol{\phi}_i)}{\sum_k h_k(\boldsymbol{\phi}_i)}, \ldots, \frac{h_{K-1}(\boldsymbol{\phi}_i)}{\sum_k h_k(\boldsymbol{\phi}_i)}\right],$$

is the entropy associated with the ith unlabeled case. This minimization problem can be solved by a gradient-descent algorithm. Derivation of the gradients is straightforward and omitted.

In the application to the real TEM image shown on the right panel of Fig. 1, we asked domain experts to determine the number of shape groups and manually pick 10 particles from each shape group to form the labeled data. According to the experts, it is sufficient to distinguish a nanoparticle into one of the four shapes: triangles, rectangles, circles, and rods. These four shapes were assigned labels 0–3, respectively. Figure 6 presents the results of applying our semi-supervised learning procedure.

6. Dealing with incomplete and composite boundaries

Real TEM images do not often provide enough evidence to obtain full particle boundaries. One cause of such situations is that a part of the boundary cannot be detected due to lack of contrast with the background, so that the detected boundary has a missing part, forming an incomplete boundary (see the first row in Fig. 7). Another cause of such situations is that multiple particles form clusters. In this case, the boundaries from several nanoparticles overlap with one another, forming a composite boundary (see the second through to the last row in Fig 7).

This section describes how to infer the full boundaries from incomplete and composite boundaries. We first propose in Section 6.1 a convexity analysis approach to split a composite boundary into individual boundaries (of single particles), each of which becomes an incomplete boundary. Next, in Section 6.2, we discuss classification of particles with incomplete boundaries. Finally, we present a shape recovery method in Section 6.3 that estimates the missing part of an incomplete boundary.

6.1. *Convexity analysis for splitting touching particles*

As we discussed in Section 3, the shapes of nanoparticles that are of concern in our applications are mostly convex. On the other hand, we notice that a composite boundary formed by multiple touching particles usually has a nonconvex shape. A convex composite boundary only occurs in the rare case of severely overlapped particles, and it is naturally difficult, even for a domain expert, to tell for whether it is one big particle or a composition of multiple overlapping particles. Thus, we focus on splitting the boundaries of the touching particles that satisfy the above non-convexity condition.

The first step is to get the smallest convex hull covering a composite boundary. We used the `Qhull` algorithm, which is a fast algorithm to find a convex hull (Barber *et al.*, 1996). Convex hulls of a few examples of composite boundaries are illustrated in Fig. 8.

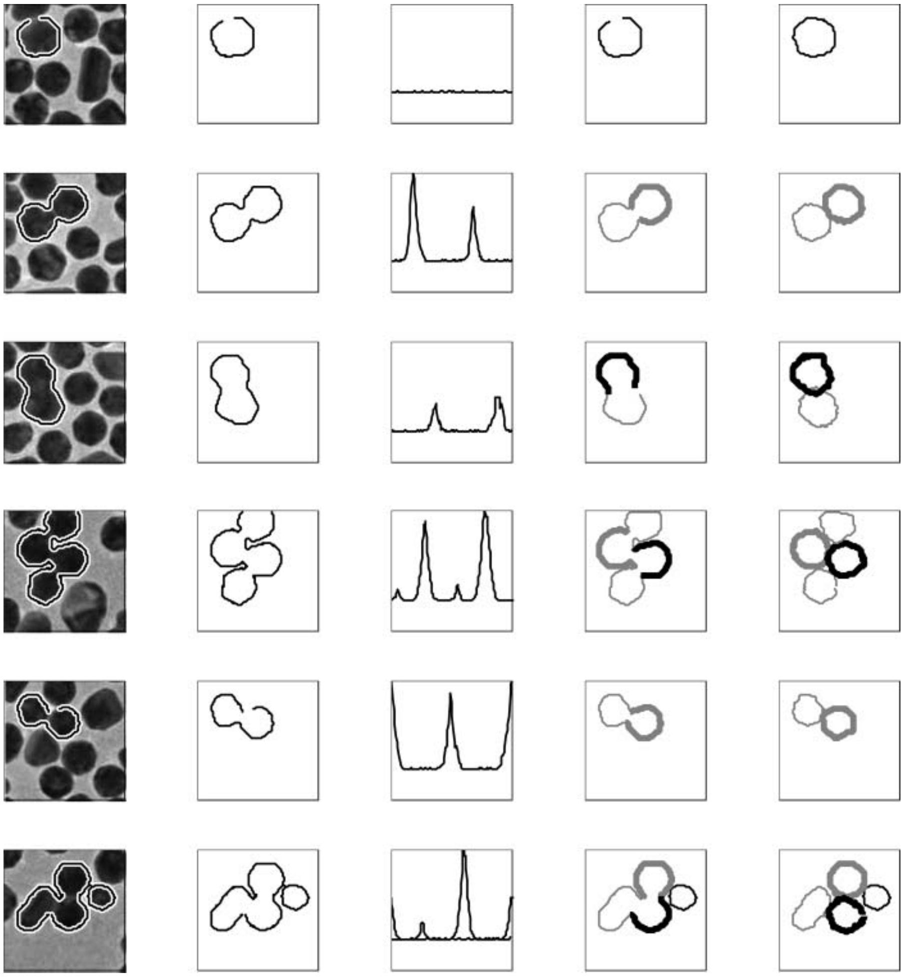

Fig. 7. Examples where an edge detection algorithm fails to extract the complete, enclosed boundaries. The first row: there are missing parts in the boundary detected. The second through last rows: composite boundaries resulting from overlapping or touching nanoparticles. From the first to last columns: the original images, boundaries detected by an edge detector, a function representing the shortest distance from each boundary point to the corresponding smallest convex hull, results of splitting the composite boundary to individual boundaries, recovered boundary.

In the next step, we find the shortest distance from each point on a composite boundary to its corresponding convex hull. The shortest distance can be viewed as a function of the polar angle representing the boundary points; see the third column in Fig. 7. It is not difficult to see that the intersecting points between individual boundaries in a composite correspond to the local maxima of the function. This understanding suggests that we only need to find the local maxima of the distance function in order to identify the splitting points for partitioning a composite boundary into a number of incomplete boundaries of individual particles. To locate the local maxima of the distance function, we first use wavelet smoothing to remove noises and then list potential local maxima by finding downward zero-crossing of the first derivative of the smoothed curve (Yang *et al.*, 2009). When two local maxima are too close, we compare their values and regard only the bigger one as a local maximum. Some examples of applying the splitting procedure are given in the fourth column of Fig. 7.

6.2. *Classification of particles with incomplete boundaries*

Incomplete boundaries arise either because of lack of background contrast or as an outcome of splitting a composite boundary. Subsequently, we need to classify the incomplete boundaries and also fill in the unobserved parts. We adopt a heuristic approach here: we first classify the particles with incomplete boundaries and then fill in the missing part of a boundary using the classification result.

We propose to use a k-nearest neighbor (k-NN) classifier to classify a nanoparticle with incomplete boundary information. We first have a rough estimate of the gravity center for the nanoparticle by taking the center of a circle fitted with the non-missing part, where the circle minimizes the

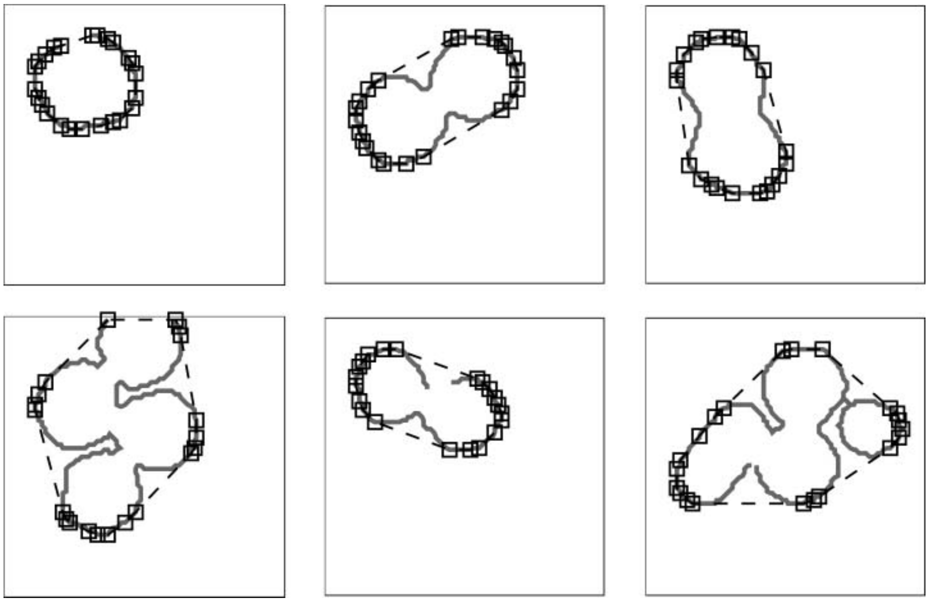

Fig. 8. Convex hulls of the composite boundaries: the dashed lines with rectangular markers are convex hulls and the rectangular markers are the extreme points of the convex hulls.

mean squared distance between itself and the non-missing part. With this estimated gravity center, the method in Section 3 can be used to yield a partially observed parametric curve representing the incomplete particle boundary. We would like to remark that our gravity center estimate would be inaccurate if the missing part is significant and such inaccuracy would have a big impact on subsequent analysis. A more sophisticated method needs to be developed for such situations, which we reserve for future research. In our study, we apply our method only to those particle boundaries whose missing part, if there is any, is no more than 20% of the entire boundary, measured in terms of polar angles.

To apply a k-NN classifier, we need to define a distance measure to identify neighbors. For a completely observed curve $\mathbf{f}_i \in \mathcal{R}^n$ in a training dataset and the partially observed curve $\mathbf{f}_* \in \mathcal{R}^s$, a distance is defined as

$$d^*(\mathbf{f}_i, \mathbf{f}_*) = \min_{t=1,\ldots,n} \|\mathbf{f}_* - \mathbf{f}_i(t, s)\|, \qquad (10)$$

where $\mathbf{f}_i(t, s)$ is a circularly completed subpart of $\mathbf{f}_i$ starting from t and having length s. That is, the distance is the minimum of the distances between $\mathbf{f}_*$ and all possible continuous subparts of $\mathbf{f}_i$ with the same length as $\mathbf{f}_*$. Based on d^*, we select k-nearest neighbors of $\mathbf{f}_*$ among all curves in the training dataset and then estimate the shape label of $\mathbf{f}_*$ by a majority vote of these neighbors. The neighborhood size k can be determined by a 10-fold cross-validation.

6.3. *Recovering the missing part of the boundary*

We borrow information from the complete boundaries in the same shape group to recover the missing part of an incomplete boundary. The classification result from the previous subsection (Section 6.2) can be used to decide on which shape group to use. We propose the following procedure. First, apply the method of FPCA proposed by Huang *et al.* (2008) to summarize the variations of shapes in a given shape group. In particular, use the complete boundaries in the same shape group to learn the principal component basis functions. Next, use the observed part of the incomplete boundary to obtain the corresponding principal component scores. Finally, combine the principal component basis functions and the principal component scores to fill in the missing part.

Specifically, we use the parametric curve representation of particle boundaries. Let $\mathbf{f}^* = (\mathbf{f}^{*t}_{\text{obs}}, \mathbf{f}^{*t}_{\text{mis}})^t \in \mathcal{R}^n$ be a partially observed curve, where $\mathbf{f}^*_{\text{obs}} \in \mathcal{R}^s$ and $\mathbf{f}^*_{\text{mis}} \in \mathcal{R}^{n-s}$ denote respectively its observed and unobserved parts. Let $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_r \in \mathcal{R}^n$ be a sample of completely observed curves, in the same shape group as $\mathbf{f}^*$, and let $\bar{\mathbf{f}}$ be the sample mean of these curves. Consider the following expansion:

$$\mathbf{f}_i = \bar{\mathbf{f}} + \mathbf{v}_1 u_{1i} + \cdots + \mathbf{v}_k u_{ki}, \qquad i = 1, \ldots, r, \qquad (11)$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_k$ are the principal component curves and $u_{1i}, \ldots, u_{ki}$ are the corresponding scores. The principal component curves are obtained by sequentially minimizing a regularized least squares criterion that penalizes the roughness of the curves (Huang *et al.*, 2008). In particular:

$$\mathbf{v}_1 = \arg\max_{\mathbf{v}} \left\{ \sum_{i=1}^{r} \|\mathbf{f}_i - \bar{\mathbf{f}} - \mathbf{v}_1 u_{1i}\|^2 + \alpha \sum_{i=1}^{r} u_{1i}^2\, \mathbf{v}^t \boldsymbol{\Omega} \mathbf{v} \right\},$$

where $\boldsymbol{\Omega}$ is a penalty matrix and α is a penalty parameter. Subsequent principal component curves are obtained similarly by using the residuals after removing preceding components. Following Huang *et al.* (2008), we use generalized cross-validation to select the penalty parameter α and we choose the number of principal components so that the majority (e.g. 99%) of the sample variation is accounted for. The reason for using roughness penalties is to guarantee that the recovered particle boundaries are smooth. Note that Equation (11) can be rewritten in a matrix form as $\mathbf{f}_i = \bar{\mathbf{f}} + \mathbf{V}\mathbf{u}_i$ where $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_k)$ and $\mathbf{u}_i = (u_{1i}, \ldots, u_{ki})^{\mathrm{t}}$. Similar to what we did in Equation (7), by partitioning $\bar{\mathbf{f}}$ and $\mathbf{V}$, the partially observed curve has the expansion:

$$\begin{bmatrix} \mathbf{f}^*_{\mathrm{obs}} \\ \mathbf{f}^*_{\mathrm{mis}} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{f}}_{\mathrm{obs}} \\ \bar{\mathbf{f}}_{\mathrm{mis}} \end{bmatrix} + \begin{bmatrix} \mathbf{V}_{\mathrm{obs}} \\ \mathbf{V}_{\mathrm{mis}} \end{bmatrix} \mathbf{u}^*. \tag{12}$$

We run a regression using the first part (i.e., the *obs* part) of the system to obtain the vector of principal score $\mathbf{u}^*$ and plug it into the second part (i.e., the *mis* part) to get the missing part of the curve. See the last column in Fig. 7 for some examples of the recovered boundaries.

After recovering the unobserved part of a boundary, the complete boundary can be used as an input to the learning method of Section 5 to reclassify the particle. We find that such a reclassification step is unnecessary and doing so rarely produces a different result. This is expected because the recovering step is based upon pre-classification of shape category information, meaning that the missing boundary is recovered based on the understanding that the complete boundary is, for instance, a circle shape. With the recovered boundary, the whole boundary will always re-enforce the belonging of that boundary to the shape category it was initially classified (i.e., the circle shape in this instance).

7. Obtaining size and shape distribution

The final destination of the statistical procedure proposed in Sections 3 to 6 is to obtain the summary statistics of the

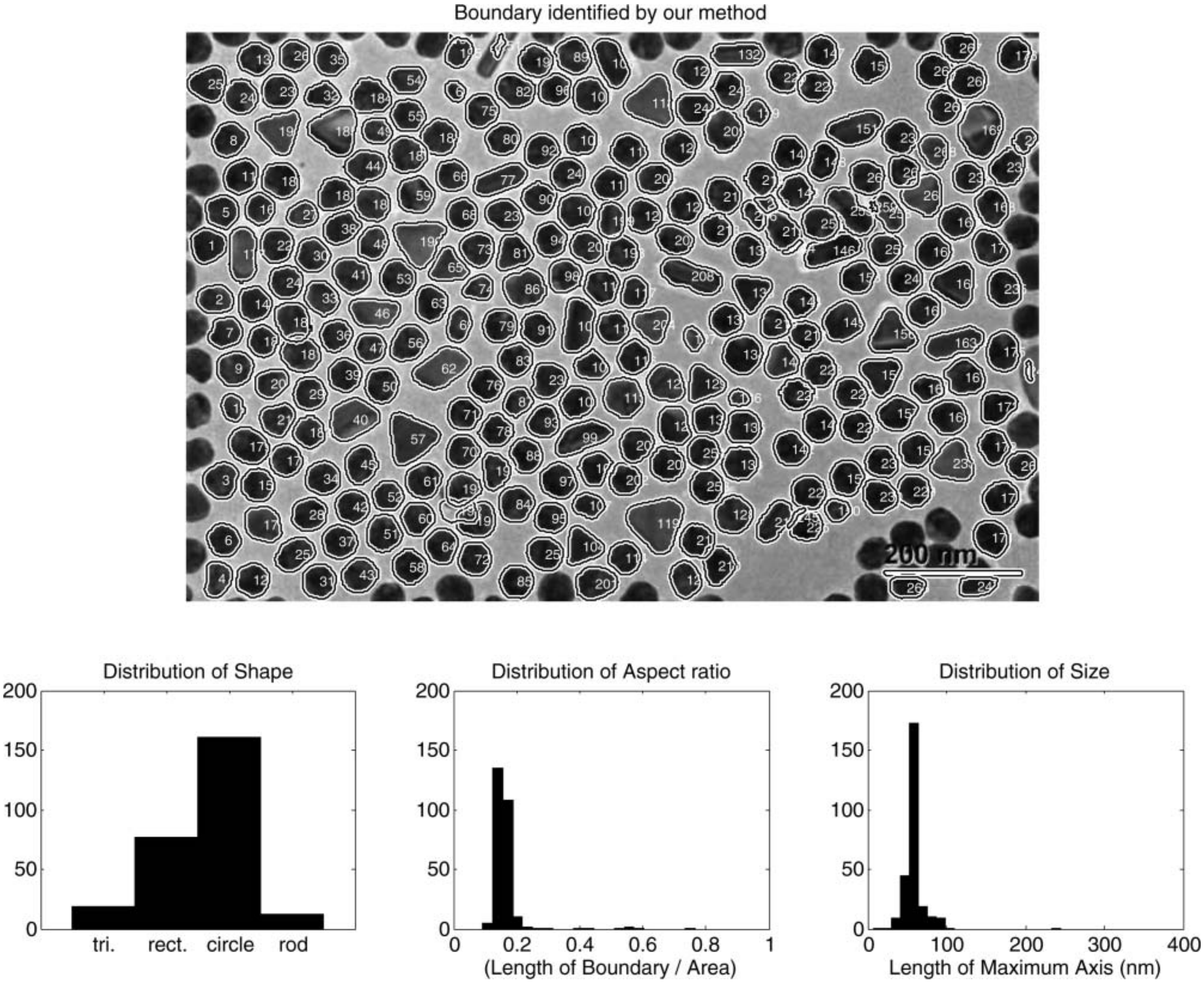

Fig. 9. Recognition results from our method for Image 4: 259 particles manually identified and 246 recognized by our method, recognition rate = 94.8%. The top figure shows the boundaries of the recognized ones.

morphology of nanoparticles, which usually includes three major distributions: (i) the size distribution (the size of a particle is characterized by the length of the longest axis of the corresponding boundary); (ii) the shape distribution; and (iii) the distribution of the aspect ratios, defined as the length of the perimeter of a boundary divided by the area of the same boundary. The three statistics are widely adopted in nanoscience and engineering to characterize the morphology of nanoparticles and are believed to strongly affect the physical or chemical properties of the nanoparticles (El-Sayed, 2001; Nyiro-Kosa *et al.*, 2009). For example, the aspect ratio is considered an important parameter relevant to certain macro-level material properties because physical and chemical reactions are believed to frequently occur on the surface of molecules so that as the aspect ratio of a nanoparticle gets larger, those reactions are more active.

Now we report the results of applying our proposed procedure to six actual TEM images under different scales. One image consists of palladium (Pd) nanoparticles prepared by microwaving a palladium solution with a surfactant. The remaining five images contain gold (Au) nanoparticles reduced from a gold salt solution heated and stirred, adding different ratios of citrate concentration. Figures 9 and 10 are two examples, where the distributions of size, shape, and aspect ratio are displayed in histograms. The results from the other images are omitted for succinct presentation of the article.

Those particles that are successfully recognized and classified are labeled by an integer number in the image. One can observe that our procedure recognizes the majority. Most of the unlabeled particles are those located on the border of the image, of which a significant portion was not observed. Domain experts deem that these particles do not need to be recognized so that they are intentionally removed before our analysis. Our classification results of particle type was also verified by domain experts and

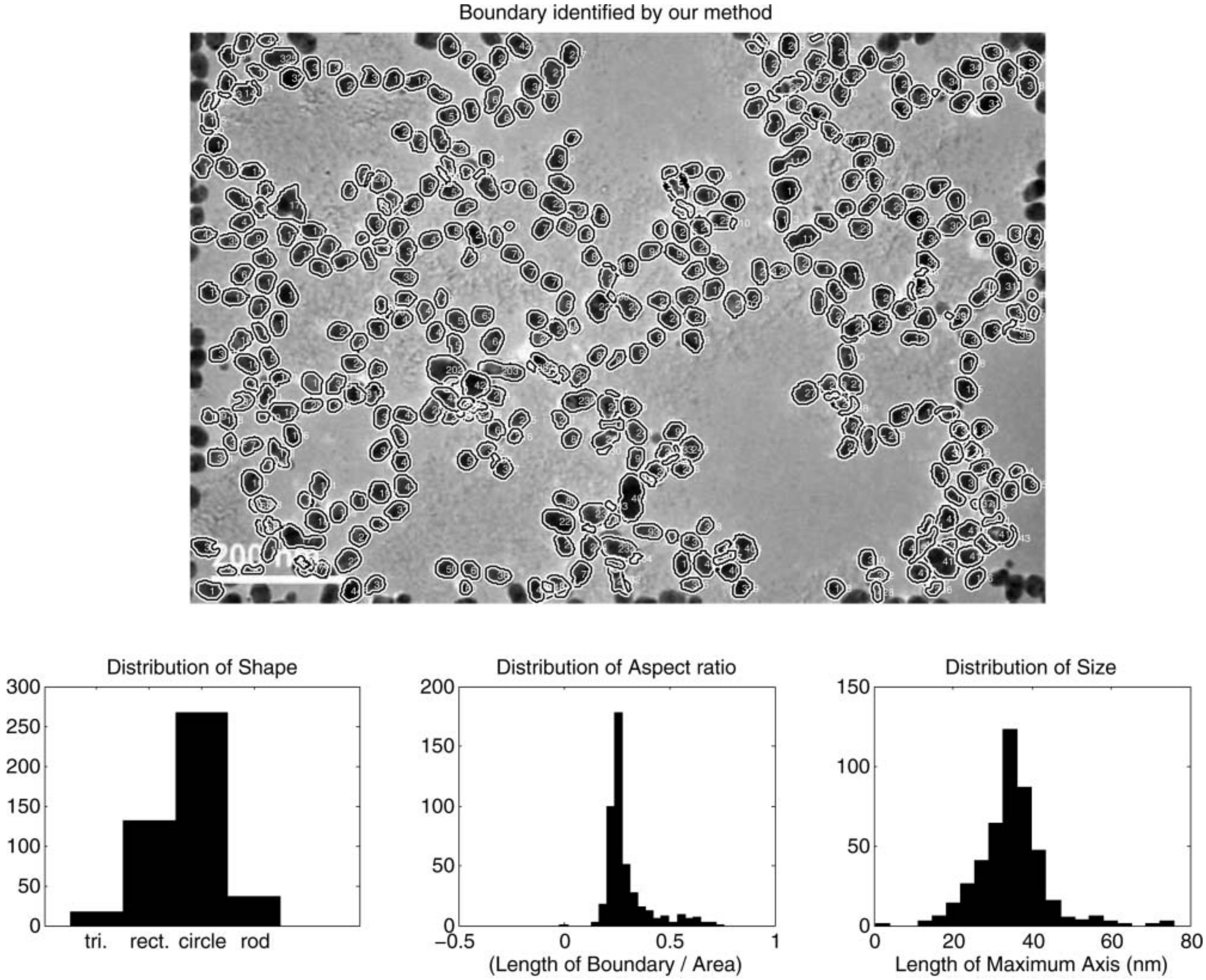

Fig. 10. Recognition results from our method for Image 6: 396 particles manually identified and 351 recognized by our method, recognition rate = 88.6%. The top figure shows the boundaries of the recognized ones.

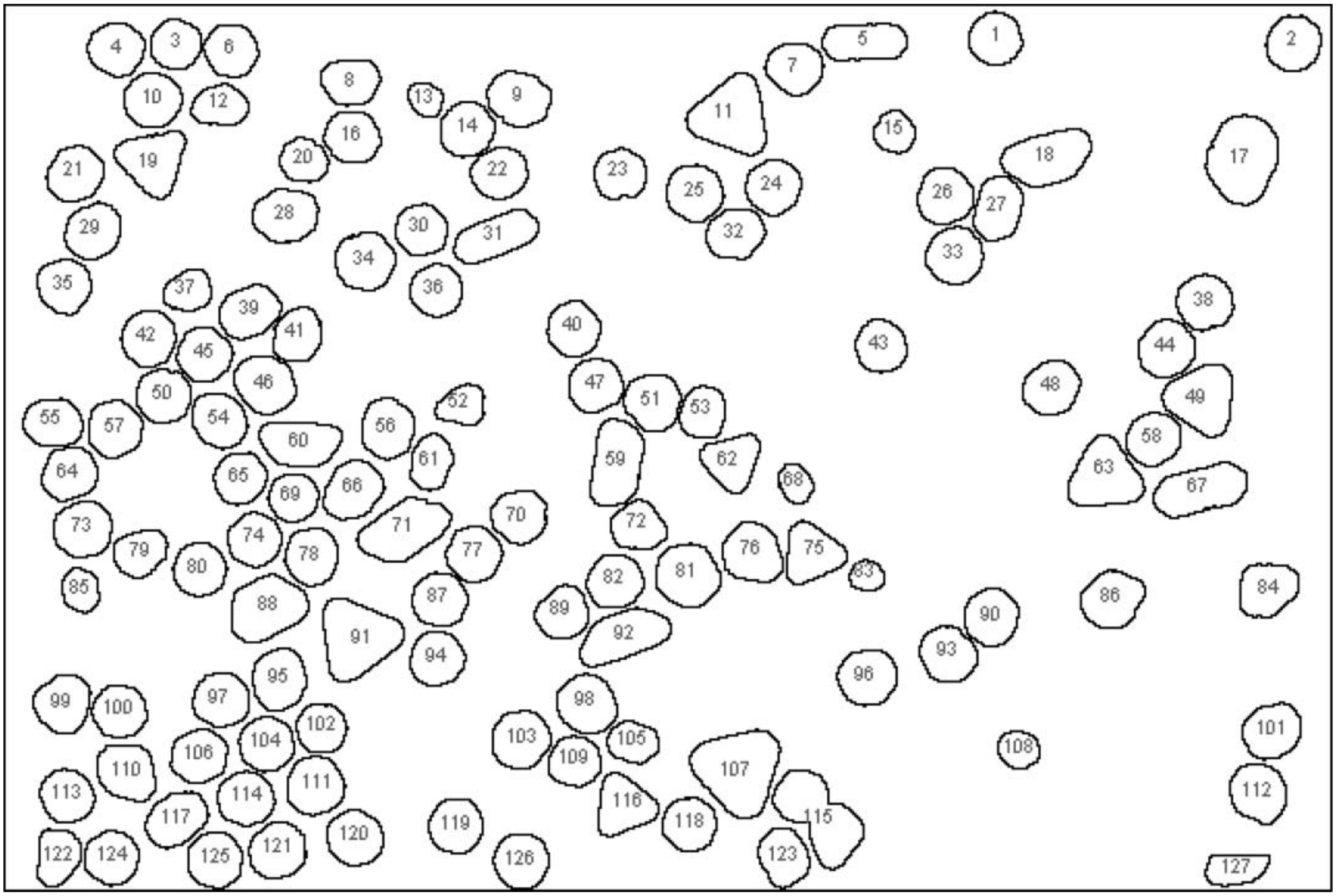

Fig. 11. Recognition results from ImageJ for Image 4: Only the recognized particles are shown. Out of the 259 particles, 124 are recognized. Recognition rate = 47.8%.

deemed satisfactorily accurate. This verification was done manually by the domain experts, who looked through each image, compared the recognition result with the original image, and then counted the number of correctly and incorrectly recognized subjects. This manual verification appears the only valid way for the time being.

As part of the verification process, we compared the accuracy of our method with the imaging tool ImageJ that is popularly used in nanotechnology research. Table 1 summarizes the number of nanoparticles recognized by our method and ImageJ for all six TEM images. For three TEM images with slight overlaps among particles, our procedure recognized 89–100% of the total particles, compared to 78–95% recognition rates of ImageJ. For three other TEM images where many particles are tangled with other ones (the images in Figs. 9 and 10 are two of them), ours approach 70–95% recognition rates, whereas ImageJ's recognition rates were 28–48%; see Figs. 11 and 12. Considering the frequent occurrence of overlaps in TEM images of nanoparticles, the existing software tool cannot be more than a supporting

Table 1. Comparison of performances on nanoparticle recognition.

			The number of correctly recognized ones	
Samples	*Degree of overlap*	*Total number of particles*	*Our method*	*ImageJ*
Image 1	Low	66	59	52
Image 2	Low	91	87	87
Image 3	Low	64	64	53
Image 4 (Figs. 9 and 11)	Pervasive	259	246	124
Image 5	Pervasive	41	29	17
Image 6 (Figs. 10 and 12)	Pervasive	396	351	110

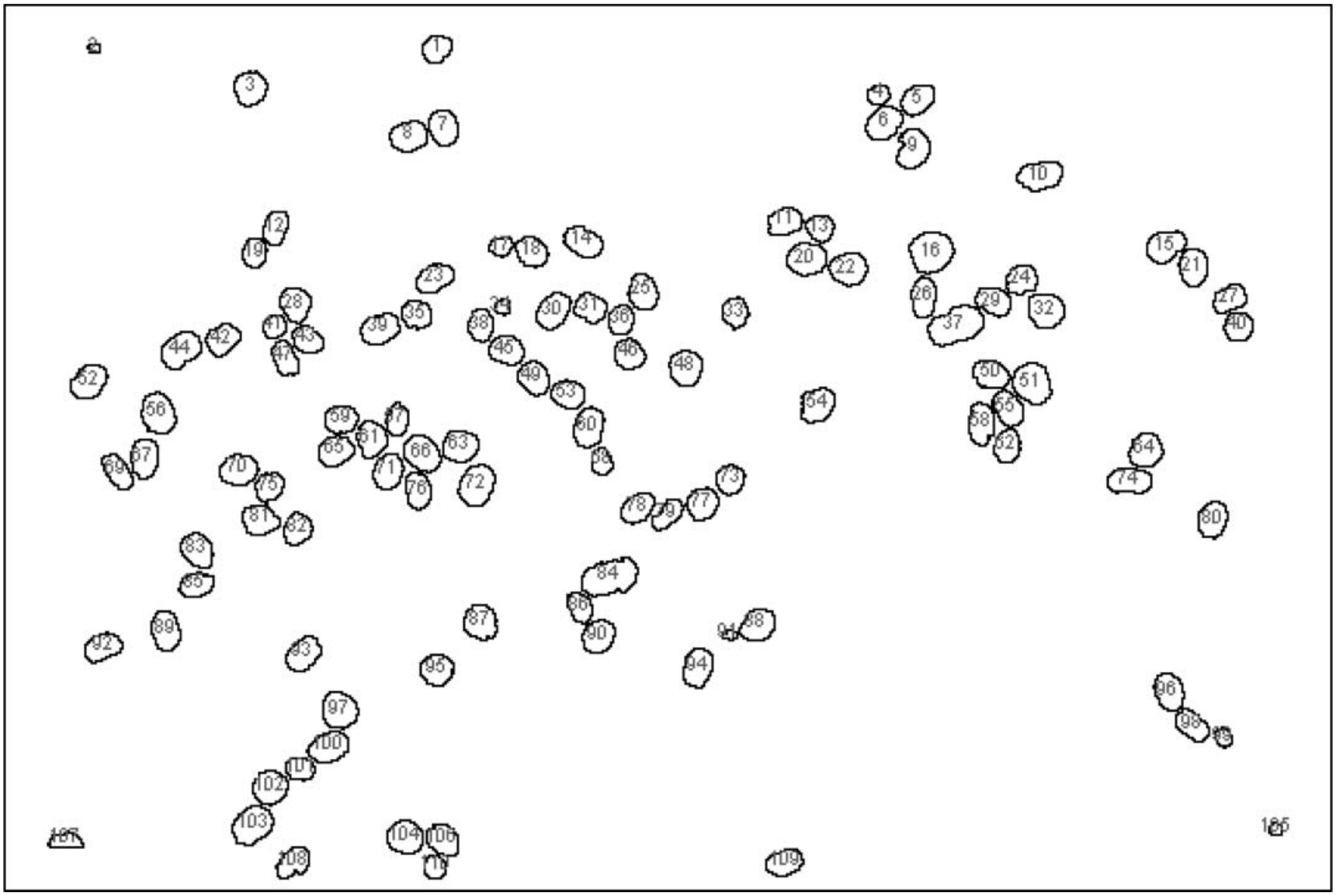

Fig. 12. Recognition results from `ImageJ` for Image 6: Only the recognized particles are shown. Out of the 396 particles, 110 are recognized. Recognition rate = 27.8%.

tool. The high recognition rate of our method can facilitate nanomaterial exploration more effectively.

8. Conclusions

Our research presents a multistage, semi-automated procedure to characterize the morphology of nanoparticles, including the following major components:

1. We use a parametric curve to characterize the morphology of nanoparticles. The curve captures the essential features of the shape of the nanoparticles and is translation and scaling invariant, which are important properties to have in order to ensure a robust shape classification.
2. We introduce the kernel matrix-based projection that projects a high-dimensional curve onto a rotation-invariant subspace of much lower dimensions.
3. We use semi-supervised learning to classify the parametric curves into a number of distinctive shape groups. Minimal degree of human expert inputs is required at this step. Otherwise, the whole procedure is rather automated. This step clusters the complete, enclosed boundaries in a TEM image and characterizes the variation in shapes within each shape group, providing the information to be used in the latter FPCA-based missing value estimation.
4. We use a convexity analysis to split composite boundaries into individual ones and recover the missing part of a boundary using the FPCA-based missing value estimation.

The merit of our development is to provide a tool for nanotechnology practitioners to recognize the majority of the nanoparticles in nano images and to obtain morphology summary statistics based on the recognized particles. We expect that our work expedites the process to quantify the morphology information of nanoparticles, which can help evaluate how well the synthesis process of nanoparticles is controlled.

A final remark is on our strategy tackling the nanoimaging problem. We employ a divide-and-conquer strategy consisting of multiple steps. As mentioned earlier, this is a result of the complicated nature of the real application. For each step we need to make choices of which statistical method to use, and our choices have been guided by the real application and our interaction with domain experts. While it is tempting to have a single statistical model that can handle the entire problem in a unified framework, to the best of our knowledge that such a unified model does not yet exist. Our experience in obtaining the current results also suggests that such a development is very challenging, because real applications present a number of difficult statistical and engineering issues that need to be addressed, including invariant, low-dimensional parameterization, non-standard data distribution on a manifold, and handling of the missing data. Improving each step of the procedure as well as developing a unified framework will require ingenious efforts from the nano-imaging community.

Acknowledgements

The authors would like to acknowledge the generous support from their sponsors. Ding and Park were partially supported by NSF grants CMMI-0348150 and CMMI-1000088; Huang was partially supported by NSF grants DMS-0606580, and DMS-0907170; Ding, Park, Mallick, and Liang were partially supported by Texas Norman Hackerman Advanced Research Program grant 010366-0024-2007; Huang, Kundu, and Mallick were partially supported by King Abdullah University of Science and Technology award KUS-CI-016-04.

References

Barber, C.B., Dobkin, D. and Huhdanpaa, H. (1996) The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, **22**(4), 469–483.

Cailliez, F. (1983) The analytical solution of the additive constant problem. *Psychometika*, **48**(2), 305–308.

Canny, J. (1986) A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**(6), 679–698.

Chen, L.-C. (2009) Investigation on morphology measurement and evaluation of TiO_2 nanoparticles synthesized by SANSS. *Journal of Alloys and Compounds*, **483**(1–2), 366–370.

Choi, H. and Choi, S. (2007) Robust kernel isomap. *Pattern Recognition*, **40**(3), 853–862.

Dryden, I. and Mardia, K. (1998) *Statistical Shape Analysis*, Volume 4. Wiley, New York, NY.

El-Sayed, M.A. (2001) Some interesting properties of metals confined in time and nanometer space of different shapes. *Accounts of Chemical Research*, **34**(4), 257–264.

Fisker, R., Carstensen, J., Hansen, M., Bødker, F. and Mørup, S. (2000) Estimation of nanoparticle size distributions by image analysis. *Journal of Nanoparticle Research*, **2**(3), 267–277.

Glotov, O. (2008) Image processing of the fractal aggregates composed of nanoparticles. *Russian Journal of Physical Chemistry A, Focus on Chemistry*, **82**(13), 2213–2218. (In English.)

Gonzalez, R.-C. and Woods, R.E. (2002) *Digital Image Processing*, third edition, Prentice Hall, Englewood Cliffs, NJ.

Huang, J., Shen, H. and Buja, A. (2008) Functional principal components analysis via penalized rank one approximation. *Electronic Journal of Statistics*, **2**, 678–695.

Jiang, S., Zhou, X., Kirchhausen, T. and Wong, S.-T.-C. (2007) Detection of molecular particles in live cells via machine learning. *Cytometry A*, **71**(8), 563–575.

Jung, M.-R., Shim, J.-H., Ko, B. and Nam, J.-Y. (2008) Automatic cell segmentation and classification using morphological features and Bayesian networks. *Proceedings of SPIE*, **6813**, 68130G.1–68130G.10.

Kothari, S., Chaudry, Q. and Wang, M. (2009) Automated cell counting and cluster segmentation using concavity detection and ellipse fitting techniques, in *Proceedings of IEEE International Symposium on Biomedical Imaging*, IEEE Press, Piscataway, NJ, pp. 795–798.

Kruskal, J.B. and Wish, M. (1978) *Multidimensional Scaling*, Sage Publications, Thousand Oaks, CA.

McFarland, A. and Van Duyne, R. (2003) Single silver nanoparticles as real-time optical sensors with zeptomole sensitivity. *Nano Letters*, **3**(8), 1057–1062.

Mohamed, M.-B., Volkov, V., Link, S. and El-Sayed, M.-A. (2000) Lightning gold nanorods: fluorescence enhancement of over a million compared to the gold metal. *Chemical Physics Letters*, **317**(6), 517–523.

Nehl, C.-L., Liao, H. and Hafner, J.-H. (2006) Optical properties of star-shaped gold nanoparticles. *Nano Letters*, **6**(4), 683–688.

Nyiro-Kosa, I., Nagy, D.-C. and Posfai, M. (2009) Size and shape control of precipitated magnetite nanoparticles. *European Journal of Mineralogy*, **21**(2), 293–302.

Pan, Y., Neuss, S., Leifert, A., Fischler, M., Wen, F., Simon, U., Schmid, G., Brandau, W. and Jahnen-Dechent, W. (2007) Size-dependent cytotoxicity of gold nanoparticles. *Small*, **3**(11), 1941–1949.

Sage, D., Neumann, F.-R., Hediger, F., Gasser, S.-M. and Unser, M. (2005) Automatic tracking of individual fluorescence particles: application to the study of chromosome dynamics. *IEEE Transactions on Image Processing*, **14**(9), 1372–1383.

Samko, O., Marshall, A. and Rosin, P. (2006) Selection of the optimal parameter value for the Isomap algorithm. *Pattern Recognition Letters*, **27**(9), 968–979.

Steinbach, M., Ertoz, L. and Kumar, V. (2003) *Challenges of Clustering High Dimensional Data*, Springer-Verlag, New York City, NY.

Tenenbaum, J.B., De Silva, V. and Langford, J.-C. (2000) A global geometric framework for nonlinear dimensionality reduction. *Science*, **290**(5500), 2319–2323.

Wang, Z.L., Petroski, J.-M., Green, T.-C. and El-Sayed, M.-A. (1998) Shape transformation and surface melting of cubic and tetrahedral platinum nanocrystals. *Journal of Physical Chemistry B*, **102**(32), 6145–6151.

Yang, C., He, Z. and Yu, W. (2009) Comparison of public peak detection algorithms for MALDI mass spectrometry data analysis. *BMC Bioinformatics*, **10**(4), 1–13.

Zhu, X. (2005) Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin–Madison, Madison, WI.

Zhu, X., Ghahramani, Z. and Lafferty, J. (2003) Semi-supervised learning using Gaussian fields and harmonic functions, in *Proceedings of the 20th International Conference on Machine Learning*, pp. 912–919.

Biographies

Chiwoo Park received his B.S. from Seoul National University and his Ph.D. degree from Texas A&M University, both in Industrial Engineering. He is currently an Assistant Professor in the Department of Industrial and Manufacturing Engineering at Florida A&M and Florida University. His research interests include data mining with its applications to nano-imaging and remote sensing. He received the best student paper award at the forth annual IEEE Conference on Automation Science and Engineering. He is a member of INFORMS and IEEE.

Jianhua Z. Huang received a B.S. in Probability and Statistics in 1989, an M.S. in Probability and Statistics in 1992 from Beijing University of China, and a Ph.D. in Statistics from University of California at Berkeley in 1997. He was on the faculty of the Department of Statistics at the University of Pennsylvania from 1997 to 2004. He is currently a Professor in the Department of Statistics at Texas A&M University and an Adjunct Professor in the Department of Biostatistics at MD Anderson Cancer Center. His research interests include computational statistics, statistical machine learning, and statistical applications in business and engineering. He is a member of ASA, ICSA, and IMS.

David Huitink received both his B.S. and M.S. in Mechanical Engineering from Texas A&M University in 2006 and 2007, respectively. As an NSF Graduate Merit Fellow, he is currently nearing the completion of his Ph.D. in Mechanical Engineering with an emphasis in Materials Science, also from Texas A&M University. His research is focused on the synthesis and fabrication of nanomaterials through novel mechanical processes.

Subrata Kundu is a research associate in the Department of Mechanical Engineering, Texas A&M University. He received a Ph.D. at India Institute of Technology. Dr. Kundu has been actively involved in synthesis of nanoparticles.

Bani K. Mallick is a University Distinguished Professor and Professor of Statistics at Texas A&M University. He holds B.S. and M.S. degrees from Calcutta University. He received his Ph.D. (1994) in Statistics from the University of Connecticut, Storrs. His research interest is in Bayesian modeling and computation. He is the author of two books, four edited books, and over 100 refereed journal publications and was named an elected fellow of the American Statistical Association, Institute of Mathematical Statistics, and the Royal Statistical Society.

Hong Liang is a professor of Mechanical Engineering, Texas A&M University. Her research focuses on nanostructured materials. Dr. Liang is a fellow of the American Society of Mechanical Engineers and the Society of Tribologists and Lubrication Engineers.

Yu Ding received a B.S. in Precision Engineering from the University of Science and Technology of China in 1993; an M.S. in Precision Instruments from Tsinghua University, China, in 1996; an M.S. in Mechanical Engineering from the Pennsylvania State University in 1998; and a Ph.D. in Mechanical Engineering from the University of Michigan in 2001. He is currently an Associate Professor and Holder of the Centerpoint Energy Career Development Professorship in the Department of Industrial and Systems Engineering at Texas A&M University. His research interests are in the area of quality and reliability engineering and systems informatics. He currently serves as a Department Editor of *IIE Transactions* and is a member of IIE, INFORMS, ASME, and a senior member of IEEE.