



Contents lists available at ScienceDirect



Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Automatic segmentation of granular objects in images: Combining local density clustering and gradient-barrier watershed

Huiguang Yang*, Narendra Ahuja

Department of Electrical and Computer Engineering, University of Illinois, Urbana, IL 61801, USA

ARTICLE INFO

Article history:

Received 8 February 2013

Received in revised form

26 September 2013

Accepted 6 November 2013

Keywords:

Image segmentation
 Granular object recognition
 Local density clustering
 Gradient-barrier watershed
 Intensity moment
 Edge detection
 Cell/nuclei segmentation
 Nanoparticle segmentation

ABSTRACT

Blob or granular object recognition is an image processing task with a rich application background, ranging from cell/nuclei segmentation in biology to nanoparticle recognition in physics. In this study, we establish a new and comprehensive framework for granular object recognition. *Local density clustering* and *connected component analysis* constitute the first stage. To separate overlapping objects, we further propose a modified watershed approach called the *gradient-barrier watershed*, which better incorporates intensity gradient information into the geometrical watershed framework. We also revise the marker-finding procedure to incorporate a clustering step on all the markers initially found, potentially grouping multiple markers within the same object. The gradient-barrier watershed is then conducted based on those markers, and the intensity gradient in the image directly guides the water flow during the flooding process. We also propose an important scheme for edge detection and fore/background separation called the *intensity moment approach*. Experimental results for a wide variety of objects in different disciplines – including cell/nuclei images, biological colony images, and nanoparticle images – demonstrate the effectiveness of the proposed framework.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Recognition and segmentation of blob or granular objects in the image is an important and fundamental task in image processing. This problem has a rich practical background in applications, such as the recognition of biological cells [12,17], cell nuclei [8–10], colonies, and pollen [34,35], as well as nanoparticles [6], and so on. Very large numbers of objects in the image make manual segmentation and counting quite tedious, if it is even feasible, so computer vision is crucial to the task. Given that objects may also vary in shape, size, and intensity and may overlap or cluster, the challenges of recognition and segmentation are in no way trivial.

The recognition of blob objects in an image can be first regarded as detecting clusters of high-density foreground pixel (pixel-of-interest) clouds in the image. For detecting the clusters of pixel-of-interest, local density clustering together with connected component analysis constitutes a good scheme and will be discussed in detail in Section 3. Local density clustering is able to cluster objects of any shape and any size. However, the major drawback of this method is that it cannot separate overlapping objects. All the clusters or objects that overlap will be grouped into the same cluster, since they are connected. To overcome this limitation, more processing is needed, such as making use of clues

in the object shape or intensity gradient within a connected component.

There have been many approaches to separating overlapping objects. These include the *watershed algorithm* [7–15], the *gradient or edge detection method* [39], *morphological erosion* [6], the *active contour method* [16–24], the *sliding band filter approach* [25,26], and others. A nice review and comments on some of these approaches can be found in [6]. The gradient or edge method apparently does not work well in cases where there is no obvious intensity difference between the overlapping objects or if the objects are strongly textured. The active contour method is quite computationally demanding, making it unsuitable for a case in which the number of objects is large, which is in fact the most meaningful case for computer-aided segmentation. The sliding band filter approach requires that the range of object size be known beforehand, and it does not work well if the size range is wide. While the watershed method is still an effective and efficient method to separate overlapping objects, improvements can be made to the algorithm.

The advantages of the watershed approach are (i) it can provide the natural growth of the region corresponding to each object independent of object shape and size, and (ii) it automatically provides a closed contour as well as computational efficiency. However, directly applying the watershed algorithm to the image or its gradient can lead to severe over-segmentation due to large numbers of local minima/maxima in the image or its gradient version. Many remedies have been proposed to overcome this issue [9,11,27–31]. Hierarchical watershed aims to

* Correspondence to: 1520 Beckman Institute, University of Illinois, 405 N. Mathews, Urbana, IL 61801, USA. Tel.: +1 217 402 5252.

E-mail addresses: hyang30@illinois.edu, hyang30@uiuc.edu, n-ahuja@illinois.edu (H. Yang).

merge the over-segmentation hierarchically to form meaningful object regions, for example based on the mosaic image transform and associated graph [27] or by multi-scale filtering of the image and segmenting on the filtered and simplified image [30]. Some other studies have proposed using the pattern classification and object model learned from the data to direct the region-merging [9,11,29]. Compared with the methods aiming to conduct a blind watershed on the image first and then merge the over-segmentations afterward, it would be better not to over-segment the image in the first place.

We believe the best way to conduct watershed segmentation is not directly from the original image or its gradient version. It is better to first find the marker corresponding to each object in the image and then to conduct the watershed based on those markers. This approach gives a much better guarantee of object counts and approximate locations in the image. Therefore *automatic detection of markers* is the most critical step in using watershed segmentation. There are several approaches to detecting markers, including the *distance transform*, *morphological erosion*, and the *gradient transform*. Under the appropriate condition, such as for convex object, the first two approaches can be shown to be essentially similar, as the final result of morphological erosion is also the local maximal distance region [6]. Both approaches are based on purely geometrical information and require the overlapping objects to display a bottleneck region as the hint for the location of separation. Both also require the individual object to be more or less convex in shape, and may lead to over-segmentation when this requirement is violated. An alternative way to detect markers is the gradient transform. It is based on the assumption that the inter-object gradient is larger than the intra-object gradient, and connected low gradient regions are detected as markers. However, this method is very sensitive to image noise and often leads to over-segmentation. Therefore using the distance transform as the basic framework and combining gradient information into the system would be a good option.

To combine gradient information into the watershed process based on the distance transform framework, one study uses the gradient-weighted distance transform [9] to alter the “distance” at a certain pixel regarding its gradient. Such a method is free of parameter tuning, but the incorporation of the gradient into the geometric framework is based on heuristics. So it is not immediately apparent where the watershed or boundary will be, or whether it will be accurate. Therefore in this study we propose an alternative version of watershed, the *gradient-barrier watershed*, in which the flooding process is still carried out based on the distance transform framework, but the image gradient directly guides the water flow in the watershed process.

In addition to the mainstream watershed techniques, we propose an important method for edge detection and foreground/background separation, which is essential for the object detection and connected component analysis. The method is based on the concept we propose in this paper called the *intensity moment*, which will be detailed in [Section 2](#).

The structure of the paper is as follows. In [Section 2](#) we discuss the concept of intensity moment and intensity moment scheme for foreground/background separation of the image. In [Section 3](#) we discuss the local density clustering method for object detection and delineation. The entire methodology and approach are discussed in detail in [Section 4](#). The experimental result is presented in [Section 5](#) and finally we conclude in [Section 6](#).

2. Foreground/background separation: the intensity moment scheme

Given the image, the first step in recognition is typically finding the foreground pixels or the pixels-of-interest. The commonly

used foreground/background pixel classification methods include *intensity thresholding* (for example, the Otsu method [32] for automatic threshold detection), the *gradient or edge detection method*, and so forth. Each method has its strengths and weaknesses, and no single method can perfectly handle all images. In this study, we propose another scheme called the *intensity moment approach*. In essence, the intensity moment approach tries to find the imbalance of the intensity distribution within the local patch of a certain scale around each pixel. In an analogy with the force moment, the intensity moment is calculated via the vector summation of the product between pixel intensity and vector distance to the patch center for each pixel within the patch ([Fig. 1](#)).

$$\vec{M}(i_0, j_0) = \sum_{(i,j) \in D} I(i,j) \vec{L}(i,j) \quad (1)$$

In above D is the local patch or domain centered at (i_0, j_0) , $I(i,j)$ and $\vec{L}(i,j)$ are the pixel intensity and the vector distance to the patch center (i_0, j_0) for each pixel (i,j) , respectively, and $\vec{M}(i_0, j_0)$ is the intensity moment of the patch centered at (i_0, j_0) .

If the intensity has variation but the overall distribution is balanced or uniform in the scale of the patch, such as is the case with local textures, then the intensity moment has only a weak response at that point. However, if there is a salient edge in the patch (the edge between the object and background), then the intensity variation is in no sense balanced, repeating, or uniform in the patch scale. Rather, there are two distinct halves in the patch, so the response of the intensity moment will be strong at that point. Therefore, by the value of the intensity moment, we can locate the salient edge between the object and the background in the image, while ignoring unwanted details. Since the intensity moment approach takes into account the balance of intensity distribution within a certain scale, it is much better than gradient edge detection at finding the salient edges or the “main structure” of the image, as shown in [Fig. 2](#). Here, what is most important is not the property of a single pixel or the gradient at that point, but rather the behavior of the local patch up to a certain scale, specifically whether it is uniform or has distinct parts. Finally, we can further classify the foreground/background pixels based on the detected object outlines.

3. Local density clustering for object detection

Besides the intensity moment approach mentioned above, some other methods can also be applied to differentiate foreground/background of the image (such as the Otsu thresholding [32]) and each method might be suitable for some case. Given we

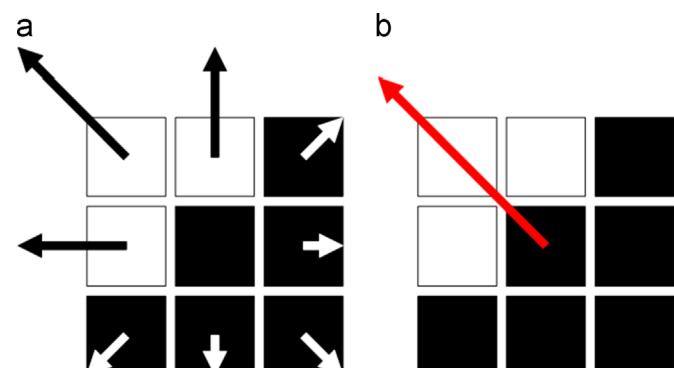


Fig. 1. Schematic illustration of the concept of intensity moment. Each block represents a pixel, dark block has lower intensity value than bright block. (a) Intensity moment and its direction of each pixel in the region with respect to the center pixel. (b) Total intensity moment and its direction of the entire region (or the center pixel).

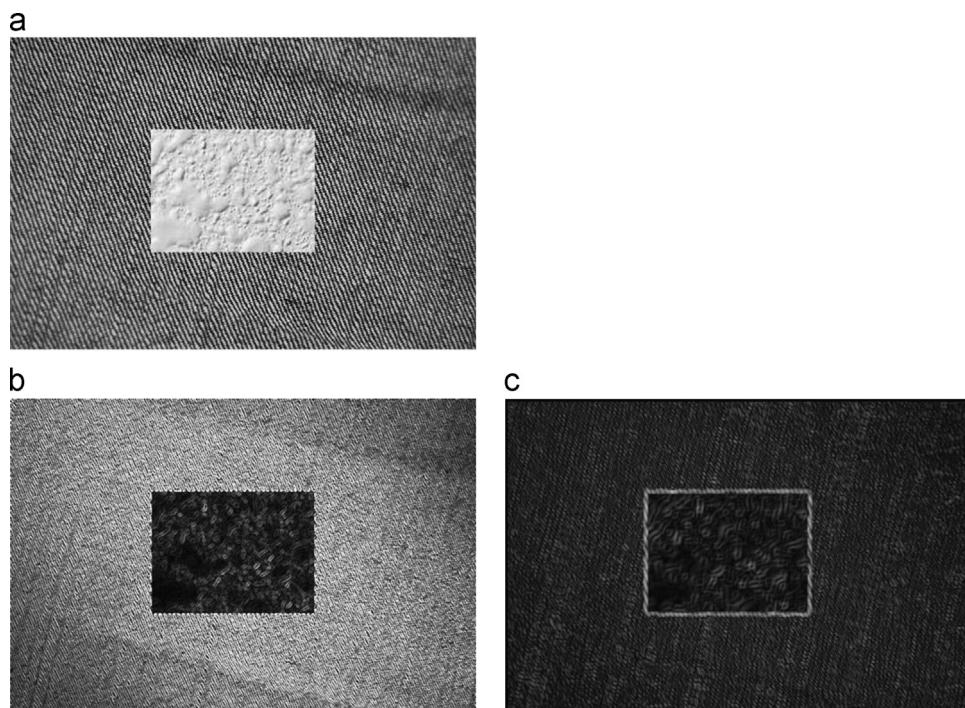


Fig. 2. Edge detection between two texture regions. (a) Original two-texture image. (b) Gradient of the original image; the boundary between two texture regions is not well detected, and the gradient value within the texture region could be higher than the value at the boundary. (c) Intensity moment of the original image; the edge between two texture regions is well detected, while the inner region response is quite weak.

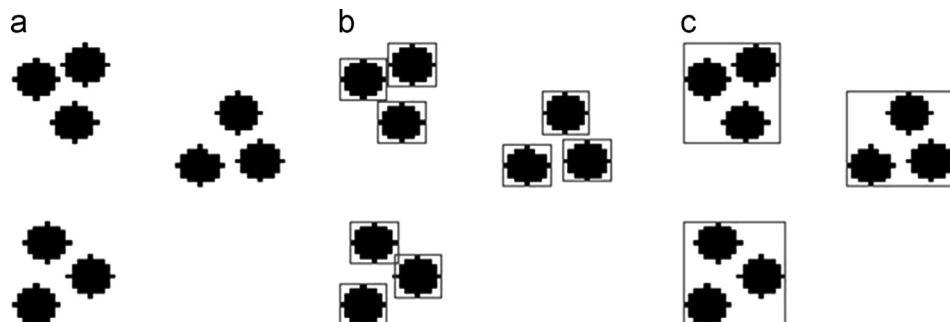


Fig. 3. The effect of scale on clustering. (a) Original image. (b) Clustering result for smaller scale (local density clustering scale = 3, local density threshold = 0.6). (c) Clustering result for larger scale (local density clustering scale = 30, local density threshold = 0.3).

already have the foreground pixels of the image, the next essential step for the recognition of granular objects is detecting the clusters of high-density foreground pixel (pixel-of-interest) clouds in the image. Therefore the objects can be detected and delineated. In cluster detection, the concept of *local density* plays an important role in the recognition task. This is because, in nature, a cluster is a group of points with relatively uniform density (typically above some density threshold) inside its domain but distinguishable from its surroundings (it is the case in this context, although the task of clustering problem has a broader definition). For a particular point in the image, we consider a local patch with appropriate scale around it. Only when the local density of the pixel-of-interest within this local patch is sufficiently high (or within a particular range in the hierarchy case) can we consider all the pixels-of-interest within this local patch to be *on-object* and in need of further clustering, while ignoring all the isolated or sparse pixels-of-interest. The merit of local density clustering is that it is independent of the cluster's size and shape and robust with respect to the individual pixel outliers. In object recognition, the property of the local region (with appropriate scale) is more important than the property of the individual pixel. It is important to note that in local density clustering the *patch scale* is not an ad

hoc parameter. It is essential for correct clustering because it reflects the perception level at which the recognition is performed. In any given case, clustering could be meaningful at various scales, so we must decide at what level we want to detect the object. Fig. 3 illustrates this concept: while clustering the image into nine objects or three objects could both be meaningful, how we cluster depends on the object scale we are interested in.

In most local density clustering methods, two input parameters are required: the *patch scale* and the *density threshold* [1]. It is not conceptually correct to expect meaningful clustering without knowing these two seemingly ad hoc parameters, since as we stressed, the clustering could be meaningful at various parameter levels and it is the recognition task itself that determines which level we are looking for. Some approaches do try to avoid manually tuning those parameters by automatically estimating the appropriate parameter settings at different levels (not eliminating these two parameters) and performing the clustering hierarchically [2–5]. However, there is probably only one parameter level in the hierarchy that best suits the clustering problem at hand.

Another essential aspect of local density clustering is *connected component analysis* [36–38]. This step follows the local density

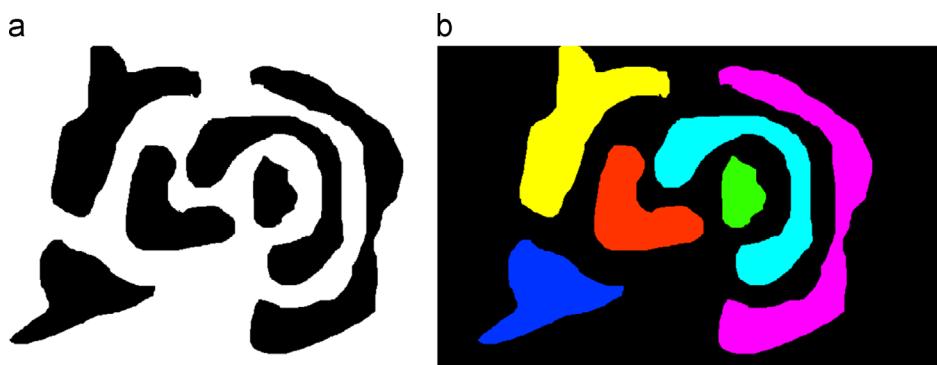


Fig. 4. Local density clustering can be applied to the clusters of any size and shape. (a) Original image. (b) Image after clustering; different clusters are shown in different colors.

test, which determines whether a given pixel-of-interest is on-object. Connected component analysis groups the on-object pixels into clusters. Here *connected* is used in the sense of “close enough”; any two on-object pixels lying within a certain distance will be considered as connected. Thus, local density clustering (with connected component analysis) is able to cluster objects of any shape and any size (see Fig. 4).

4. Methodology

4.1. Overview

As in Sections 2 and 3, we have discussed the separation of foreground/background in the image and the local density clustering for object detection and delineation. Up to this point, all the objects that are separated can be segmented successfully. In order to further separate the overlapping objects, more processing is needed, which will be discussed in detail in this section. The schematic illustration of the overall approach is shown in Fig. 5. At the beginning of this section, local density clustering and connected component analysis will be revisited briefly to make connection with the experiment carried out in this study.

4.2. Local density clustering and connected component analysis

After foreground/background pixels are classified in the image, we will evaluate the local density at each foreground pixel. That is, for a patch with a proper scale centered at the pixel, we compute the ratio of foreground pixel number to total pixel number in that patch to see if the foreground pixel density is sufficiently high locally. The patch scale should in general be smaller than the object scale but large enough for making a statistically confident decision regarding the density. We find in most cases that the proper scale could be just a few pixels and that the clustering result is not sensitive to this parameter. Actually, in most experiments we use a patch scale of 3 pixels and patch density threshold 0.8, i.e., the ratio of foreground pixel number to total pixel number in the patch is above 0.8. The optimal parameters could change if different dataset is considered. We then consider all the foreground pixels within the patch to be on-object and in need of further clustering.

All the on-object pixels within the same patch will be grouped into the same cluster. Alternatively, any two on-object pixels within the distance of the patch scale will have the same cluster label. The connected component analysis is applied to cluster all the on-object pixels into multiple connected components, each with a unique label. The local density clustering together with the connected component analysis can cluster the objects with any shape and any size, although it cannot differentiate overlapping objects.

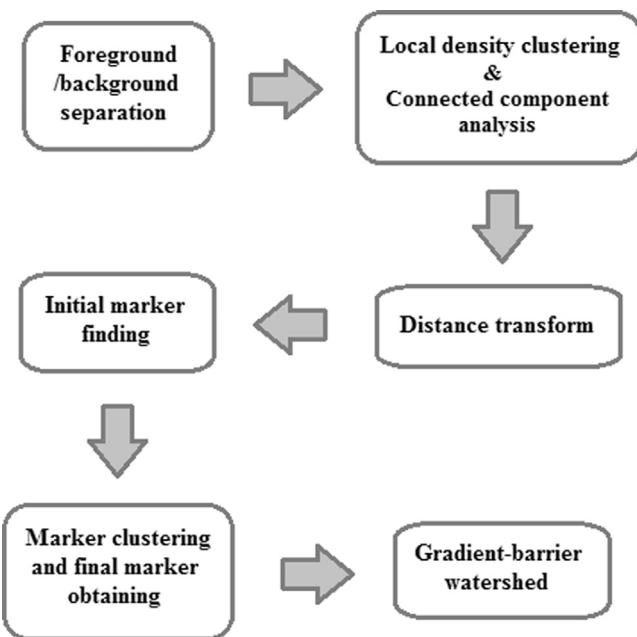


Fig. 5. The general flowchart of the approach.

4.3. Distance transform and marker finding

If we have no information on the shape or size of the object, the above procedures should already provide the best possible clustering result. However, if we have some prior knowledge of the object shape (for example, circular), then we can make a judgment as to whether there is overlapping. Up to this point, all the clusters or objects that are overlapping will be grouped into the same cluster, since they are connected. But if the objects are circular, a very elegant scheme exists to separate overlapping ones. We make use of the distance transform [40,41] or inner-distance map based on the assumption that, for a circular object, the circle center typically has the largest distance to the background compared with any other point inside the circle. When two circular objects overlap, their two centers still tend to have the local maximal distances to the background. Therefore, if we compute each on-object pixel's distance to the nearest background pixel and plot the inner-distance map, the local maxima in the map will usually correspond to the centers of the circular objects, even if they overlap. Based on those local maxima or *markers*, we find the locations of each object and can further classify each pixel into its proper cluster.

To be specific, the distance transform of a foreground pixel p is

$$D(p) = \min_{q \in B} d(p, q), \quad p \in F \quad (2)$$

where F is the set of foreground pixels and B is the set of background pixels.

To find the markers of the image or the local maxima in the inner-distance map, we check two criteria: (i) whether the current pixel itself is a local maximum, i.e., whether its pixel value is greater than or equal to any of its neighbors, and (ii) whether the patch centered at this pixel also represents a local maximum in the larger scale. The purpose of (ii) is to eliminate the impact of noisy pixels or outliers. This is done because the noise point may easily become a local maximum, but it is hard to support a patch in a larger scale to still be a local maximum. In this study, we consider a 5×5 patch centered at the each pixel and compared this patch with all its eight 5×5 neighboring patches to determine whether it is a local maximum. The sums of the pixel values in each patch are compared. In this way, the detection of the markers will be much more robust.

The above condition is described as the following, where $I(i,j)$ is the value of pixel (i,j) , in the inner-distance map, $P(i,j)$ is the sum of pixel values of the 5×5 patch centered at pixel (i,j) .

"Loop for all the pixels of the image (ignore the boundary effect)
For pixel (i,j) " are part of formula (3), not the main text, please separate it from main text and put it in the format as in my submission.

Loop for all the pixels of the image (ignore the boundary effect).
For pixel (i,j)

If $I(i,j) > \max\{I(i-1,j-1), I(i-1,j), I(i-1,j+1),$

$I(i,j+1), I(i+1,j+1), I(i+1,j),$

$I(i+1,j-1), I(i,j-1)\}$ AND

$P(i,j) > \max\{P(i-5,j-5), P(i-5,j), P(i-5,j+5),$

$P(i,j+5), P(i+5,j+5), P(i+5,j), P(i+5,j-5), P(i,j-5)\}$

Label pixel (i,j) as the marker pixel

End

4.4. Marker clustering

After all the markers are generated, we perform an additional clustering on the markers to group those that should belong to the same object into one marker. It is different from seed filtering in terms that we do not know in advance how many final marker we should end up with, therefore it requires unsupervised clustering. This is done because local maxima searching in the inner-distance map often results in multiple local maxima for one object. This is typical for elliptical objects, where the multiple local maxima usually appear as a line along the long axis of the ellipse. The clustering of the markers can be carried out in the same spirit of connected component analysis. We specify a proper scale, and we cluster every two marker points lying within this distance into the same group. Typically, this scale is much smaller than the scale of the object in the image, usually just a few pixels, and we use 10 pixels in most of our experiments. Finally, the average location of the markers in one cluster will be used as the final marker to represent this group (Fig. 6).

4.5. Final segmentation based on markers

An isolated object that has been successfully identified by connected component analysis may have no marker detected in the region. For this reason, we ignore connected components with no or only one marker inside, keep their cluster labels unchanged, and focus on connected components with multiple markers inside them to separate further.

Once we have the correct markers, we have the approximate counts and center locations of the objects in the image. The simplest way to find the boundaries between overlapping objects is to apply a *nearest-neighbor assignment*, that is, to assign each on-object pixel to its nearest marker. However, the resulting object boundary may be quite unnatural looking, and boundary locations will become

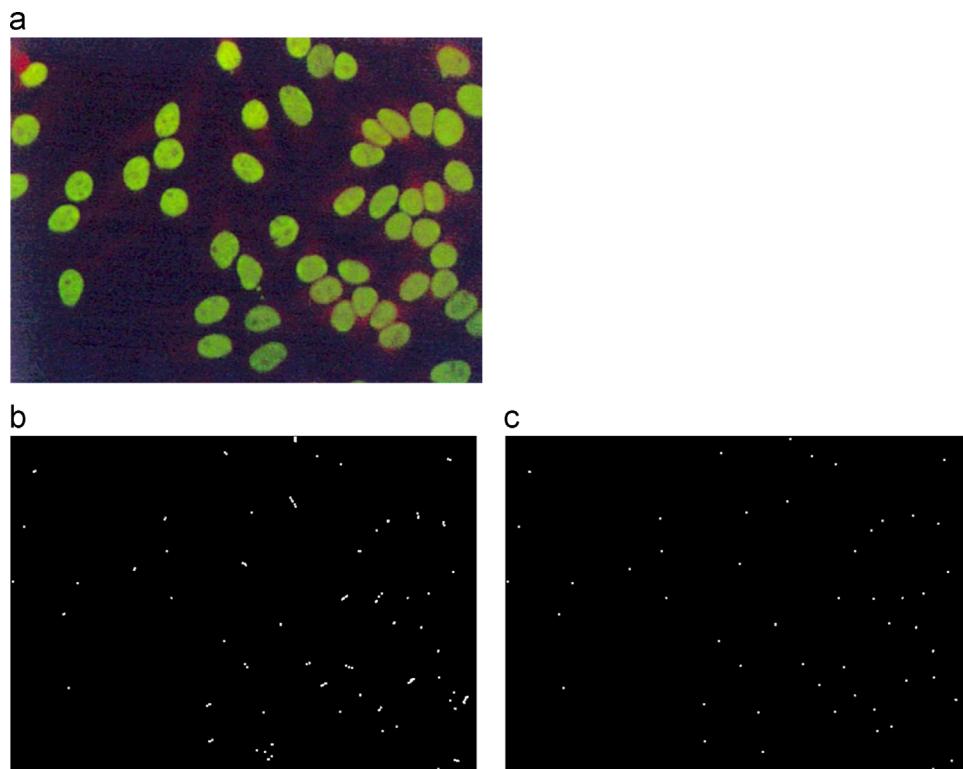


Fig. 6. (a) Original image. (b) Markers obtained from local maxima of inner-distance map; note one object may have multiple markers. (c) Markers after clustering; one object typically has one marker corresponding to it.

very inaccurate when object sizes differ much. The watershed algorithm is a very natural and effective way to find the boundaries. It can be applied based on the purely geometric information (the inner-distance map) and possibly combined with the intensity (gradient) information. In this subsection, we will first discuss the purely geometric-based watershed.

For each connected component with markers inside, we start the watershed from the marker with the maximal distance value. We decrement this value by one each time and search for all the unlabeled pixels within this connected component whose distance values are larger than this current value. For each such pixel, we check its four neighbors (top, bottom, left, and right) to see if any of them already has a label (initially, only the marker pixels have labels, and different markers will have different labels). If a neighboring pixel does have a label, we assign that label to the unlabeled pixel. At the start, only the neighbor pixels of the markers will be assigned labels. If the four neighbors have different labels, that means that this unlabeled pixel is at the edge of two or more connected objects, each centered at a corresponding marker, and we denote this pixel as the *watershed* or the *boundary*. We loop from the maximal distance value to zero (the distance value for the outermost pixels on the connected component). To find and label all the pixels above a certain distance value, we can either do a connected component analysis for all such pixels or simply do multiple-round scanning. For each round of scanning, the pixels adjacent to some already labeled pixel will be labeled, while those (if any) appearing far away and with no labeled neighbors will be left blank and wait for future scans. The image could be scanned differently each time to make sure the growth of watersheds is in balance. In most cases, we find scanning 4–6 times at each distance level is sufficient. The connected component is scanned from top-left to bottom-right or vice versa, alternatively, for each round. The algorithm is shown in Fig. 7.

4.6. Gradient-barrier watershed

The purely geometric-based watershed does not take into account the intensity information of the image, so its boundary is more or less a guess based on the shape of the region. But sometimes the boundary between overlapping objects is obvious from the image intensity or gradient, so it is best to make use of such information. Some studies tried to combine the gradient information into the distance transform framework, for example using the gradient-weighted distance transform [9] to alter the “distance” of a certain pixel regarding its gradient. This method has its merit, as discussed later, but its formulation is based on some heuristics. It is not immediately clear that incorporating gradient information into

```

Label all the markers with different labels
Label all other pixels on the current connect component with label 0

For current_distance_value = maximal_distance_value : 0 (decrement
current_distance_value by 1 each time)
    Scan the image (can do multiple times, each time with a different order)
        If pixel_distance_value > current_distance_value
            Check the 4-neighbor of the pixel
                If some of them already has a label (other than 0)
                    If they all have the same label
                        Label the current pixel with the same label
                    Elseif they have different labels
                        Label the current pixel as the watershed or boundary
                End
        End
    End
End

```

Fig. 7. The algorithm description for purely geometric-based watershed applied in this study.

the geometric framework in this way will indicate where the watershed or boundary will be, or do so accurately. Further, this method may lead to the wrong location and distribution of markers and hence to the merge or split of the object.

The gradient-barrier watershed we propose uses the image gradient to directly guide the water flow in the watershed process. The water flow will be blocked at the pixels with strong gradient rather than passing through, therefore the strong gradient regions act as a barrier for the water flow; water can only flow around them. The preliminary watershed or boundary will be determined not only at the locations where waters from different markers meet, but also at the locations where the water is blocked by those strong gradient pixels. Of course, the object may have inner intensity fluctuations or textures. If such regions are also detected as strong gradient regions, then basically water from the same marker will go around those regions. These “isolated islands” can be detected and eliminated by a filling operation, since they are surrounded by the pixels of the same cluster.

Therefore with gradient-barrier watersheds we need to first obtain the gradient image of the current overlapping region and then look for the pixels in the *inner part* of this overlapping region with a gradient greater than some threshold. This is because the gradient at the outer boundary of the overlapping region (against the background) is usually high and is not the inner boundary between the overlaps that we are looking for. Finally, all the high-gradient pixels we detect in the inner part of the connected region will become the barriers for the watershed process. The algorithm is shown in Fig. 9.

The gradient-barrier watershed is best suited for the case in which the object has relatively uniform intensity while the inter-object boundary or intensity gradient is obvious. While in the case in which the inner-object intensity fluctuates much or the inter-object boundary is not apparent, it may be better to conduct watershed based only on the geometric information.

Both the *gradient-barrier watershed* and *gradient-weighted distance transform* method have merits. In the *gradient-weighted distance transform*, there is no need to estimate the parameter of the gradient threshold in order to determine which pixels act as barriers for water flow. But its formulation is based on heuristics, such as the exponential decay structure, while the *gradient-barrier watershed* provides a more natural and explicit way to find the boundary between overlaps. Also, since the *gradient-weighted distance transform* assigns a new “distance” (altered by the gradient) to each pixel on the object, it may affect the detection and distribution of markers. The *gradient-weighted distance transform* method often leads to wrong marker location and distribution and hence to an incorrect merge or split of the object, especially if the object is strongly textured. The *gradient-barrier watershed* does not change the marker location or distribution. Therefore, if the markers have been reliably detected, the *gradient-barrier watershed* will perform more robustly and preserve correct object counts. Actually, we find the *gradient-barrier watershed* method is not sensitive to the parameter of gradient threshold. In most experiments, we use a gradient threshold of 30 for the grayscale images, which works well. In the pollen image shown in Fig. 8, we see that two pollen grains on the upper-right are merged in the *gradient-weighted distance transform* watershed result. This is due to the generation of wrong markers in that region (markers are clustered based on proximity). In contrast, the *gradient-barrier watershed* provides the correct marker distribution and gives the correct segmentation (Fig. 8).

5. Experiment

Granular object recognition has a rich application background. Here we test our scheme on biological cell/nuclei images, colony

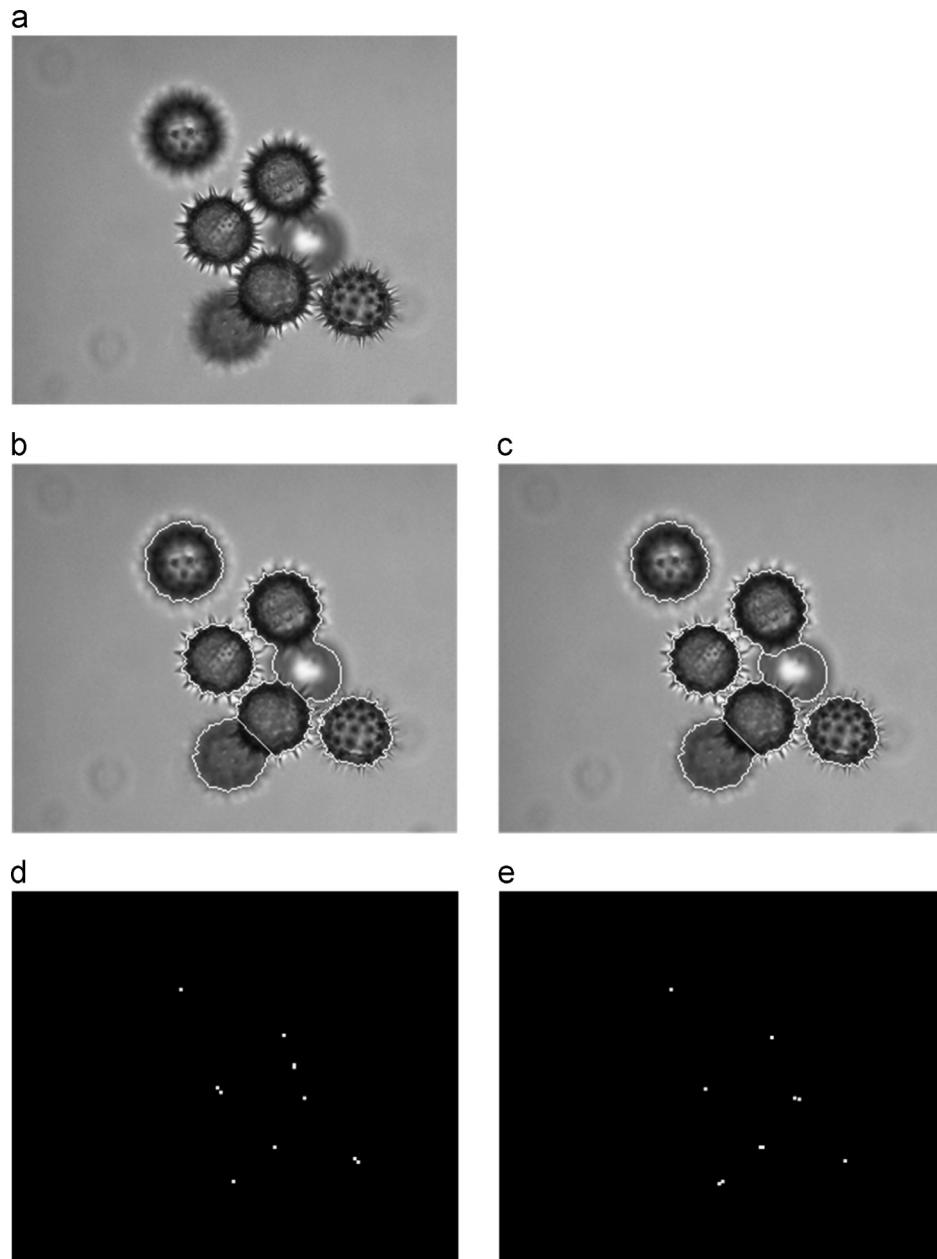


Fig. 8. Comparison between gradient-weighted distance transform watershed and gradient-barrier watershed. (a) Original pollen image. (b) Segmentation result by gradient-weighted distance transform watershed. (c) Segmentation result by gradient-barrier watershed. (d) Markers found in gradient-weighted distance transform watershed. (e) Markers found in gradient-barrier watershed.

images, pollen images, as well as physical nanoparticle images. This is only a small selection of this approach's potential applications (Figs. 10–12).

For the blood cell image in Fig. 13, we can differentiate its foreground and background simply via intensity thresholding. The local density clustering with connected component analysis can then be performed, and all the connected regions of the foreground will be clustered and labeled (Fig. 13. (b)). At this point, all the isolated objects have been successfully segmented. In order to segment the overlapping objects, we perform the distance transform on the connected component analysis result to obtain the inner-distance map of the original image (Fig. 13(c)). The markers are then located based on the inner-distance map (Fig. 13(d)). As mentioned earlier, the marker pixel should satisfy both of the following criteria: (i) it should be a local maximum pixel-wise, and (ii) the 5×5 patch centered at this pixel should be a local

maximum compared with all the eight 5×5 patches surrounding it. The markers are then clustered, and two markers within the distance of 10 pixels will be grouped into the same cluster. The average marker location in each cluster will be computed and used as the final marker representing this cluster (or object) (Fig. 13(e)). In this way, the multiple markers within one object will not lead to the splitting of the object. Then the segmentation of the overlapping regions will be carried out via gradient-barrier watershed with a gradient threshold 30, i.e., we consider all the pixels with gradient greater than 30 as barriers to the water flow.

The threshold value can be determined from the gradient of the original image by observing the general range of the gradient value at the overlapping object boundaries. Or, it can be extracted automatically by the histogram mode-seeking method. If the threshold is hard to determine, we can simply perform a purely geometric-based watershed. Also, in the algorithm we only

```

Label all the markers with different labels
Label all other pixels on the current connect component with label 0

For current_distance_value = maximal_distance_value : 0 (decrement
current_distance_value by 1 each time)
    Scan the image (can do multiple times, each time with a different order)
        If pixel_distance_value > current_distance_value
            Check the 4-neighbor of the pixel
            If some of them already has a label (other than 0)
                If they all have the same label
                    If pixel_gradient_value < threshold
                        Label the current pixel with the same label
                    Elseif pixel_gradient_value >= threshold
                        Label the current pixel as the watershed or boundary
                    End
                Elseif they have different labels
                    Label the current pixel as the watershed or boundary
                End
            End
        End
    End
End

```

Fig. 9. The algorithm description for gradient-barrier watershed; the difference with a purely geometric-based watershed is highlighted.

consider the inner part of each connected component with a distance value (in the inner-distance map) larger than 5 pixels, i.e., the pixels at least 5 pixels away from the outer boundary. This is done to avoid the influence of the high gradient at the outer boundary of the connected component.

The segmentation result of the gradient-barrier watershed is shown in Fig. 13(g). Compared with the watershed result based on the purely geometric information (Fig. 13(f)), we can see the significant improvement of boundary locations. A few exceptions where the gradient-barrier watershed also did not give an accurate boundary location are largely due to the lack of inter-object gradient within the overlapping region. Finally, a post-processing step will help to rule out a few non-cell objects based on size (Fig. 13(h)). The final count of correctly segmented cells in the image is 194, out of the ground truth 221. Note that there are a few merges, splits and missed cells in the final segmentation (Table 1). Also, the presence of the white blood cell in the image creates some trouble for the segmentation (denoted as "Interrupted" in the table), but overall the segmentation performance is good.

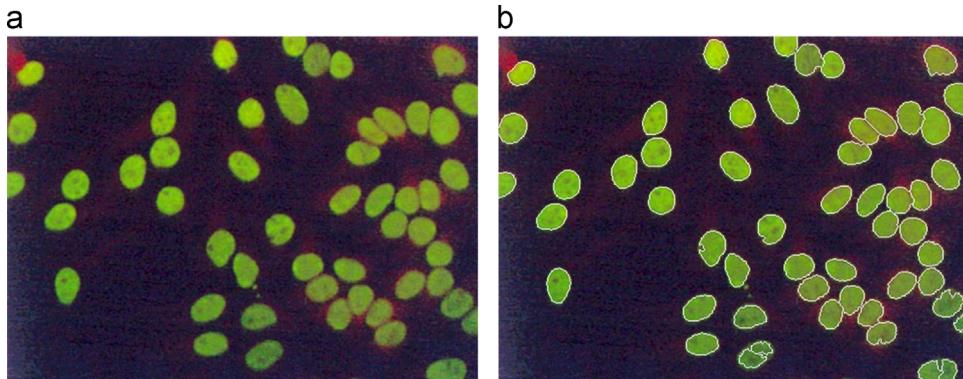


Fig. 10. (a) Cell nuclei image and (b) segmentation result.

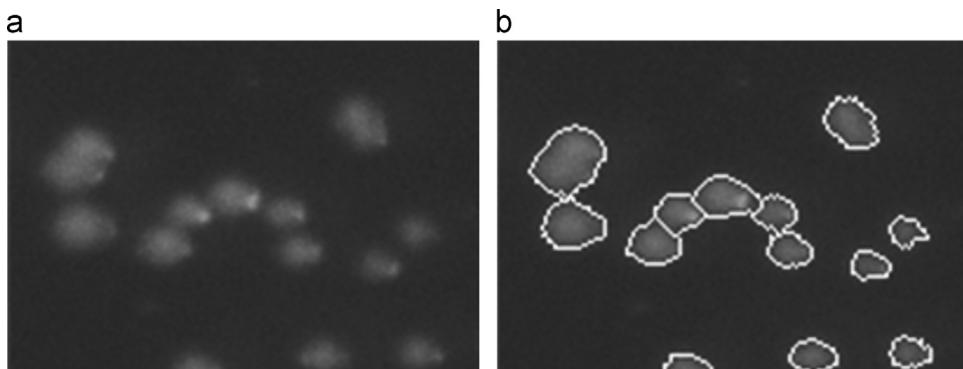


Fig. 11. (a) Colony image and (b) segmentation result.

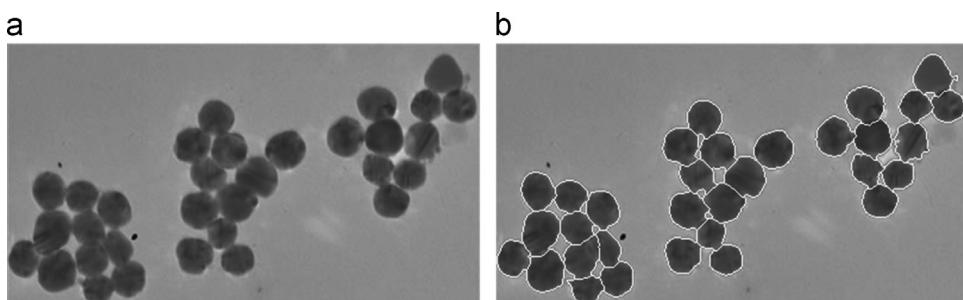


Fig. 12. (a) Nanoparticle image and (b) segmentation result.

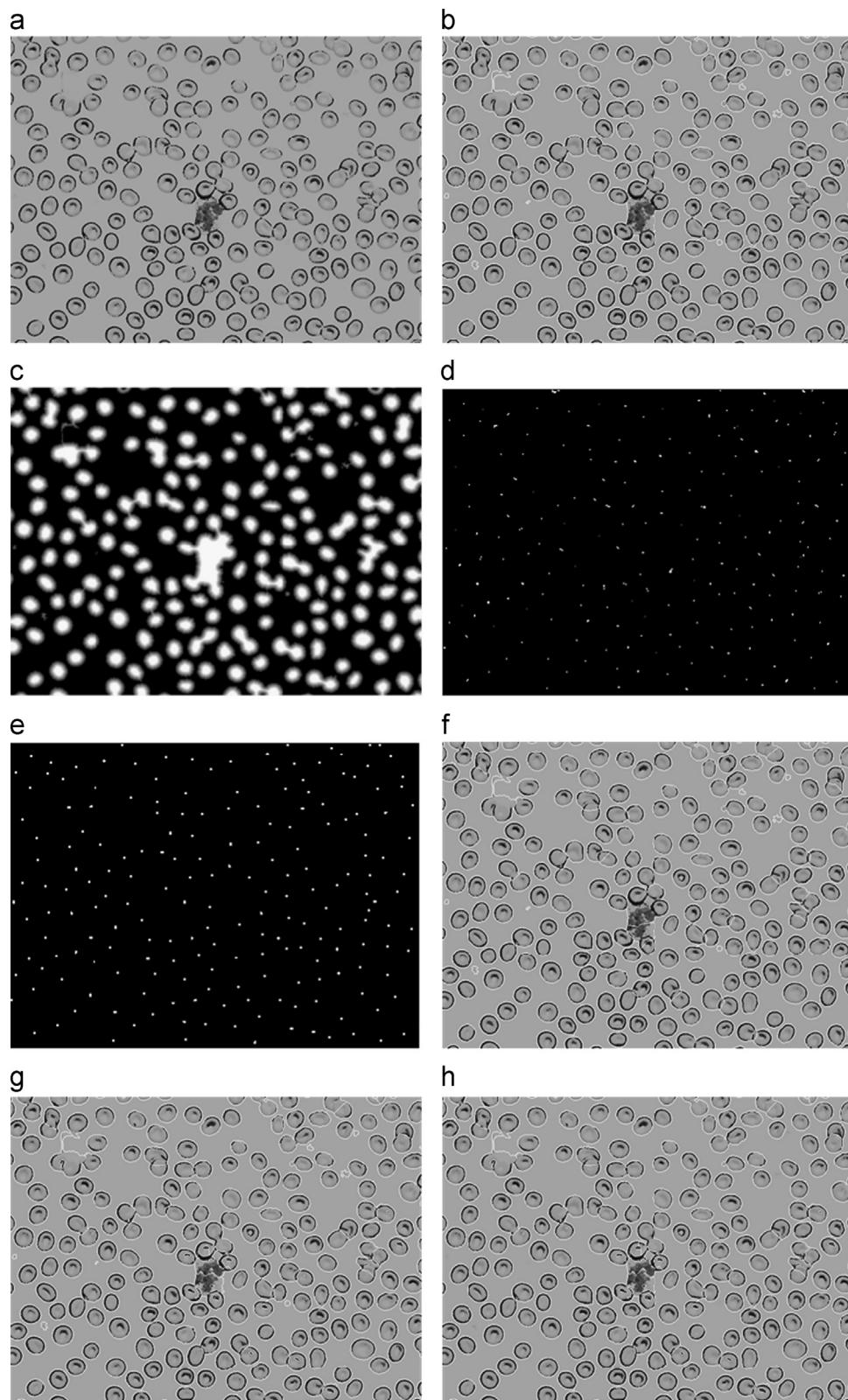


Fig. 13. (a) Original blood cell image, (b) Local density clustering with connected component analysis, (c) Inner-distance map, (d) Object markers obtained from local maxima (upon to certain scale) of inner-distance map, (e) Object markers after clustering, (f) Watershed based on purely geometrical information (markers and inner-distance map), (g) Gradient-barrier watershed based on both geometrical information (markers and inner-distance map) and intensity information (intensity gradient), (h) Gradient-barrier watershed after post-processing.

The comparison between the gradient-weighted distance transform watershed and gradient-barrier watershed (our method) is conducted on the *E. coli* image (Fig. 14). The performances of two

methods are comparable yet our method is better (Table 2). Gradient-weighted distance transform watershed is more easily to make object split or part of the object merged into the adjacent one. More

examples on cell nuclei image segmentation and *E. coli* image segmentation are shown in Fig. 16 with Table 3 and Fig. 17 with Table 4.

The same experiment is carried out on the coin image (Fig. 15). We see from the result that the gradient-weighted distance transform method splits the objects. The gradient-barrier watershed does not split the object, and keeps the object shape and correct object count in general (although the segmentation for the top-left coin is not perfect).

6. Conclusion

Granular object segmentation has fundamental importance in image processing and object recognition. It can emerge from various application backgrounds, such as cell/nuclei image segmentation in biology and nanoparticle image segmentation in physics. The more advanced recognition tasks, such as counting, classifications, need to be built on the segmentation result. The

Table 1

The segmentation performance of gradient-barrier watershed on the blood cell image.

Ground Truth	Correctly segmented	Merged	Split	Missed	Interrupted
221	194	10	1	9	7

entire task can be roughly divided into two parts: (i) the recognition of all the objects-of-interest in the image, whether they are isolated or overlapping, and (ii) the recognition of each individual object from the overlapping region. The first stage can be realized by local density clustering with connected component analysis, and the second stage can be realized by the proper marker-finding procedure and proper watershed algorithm.

It is important to mimic human visual perception in the image-recognition task, and local density clustering is the method that tries to do this. The local density of pixel-of-interest at each local patch of certain scale determines whether this patch is significant enough to become part of the object-of-interest. In most cases, it is the property of the patch that matters, not the property of the individual pixel. The scale of the patch is also a key factor in the recognition process. It reflects the perception level at which the recognition is performed and depends on the particular recognition task. The scale can be determined hierarchically, but probably only one or a few levels out of the hierarchy will best suit the particular recognition task. Local density clustering can recognize a cluster or object with any shape and size, but it cannot separate overlapping objects. The marker-finding procedure and watershed algorithm provide powerful tools for further segmenting overlapping granular objects.

We aim to establish one-to-one correspondence between the markers and objects in the image and then perform watershed upon those markers. In practice, there are usually multiple markers

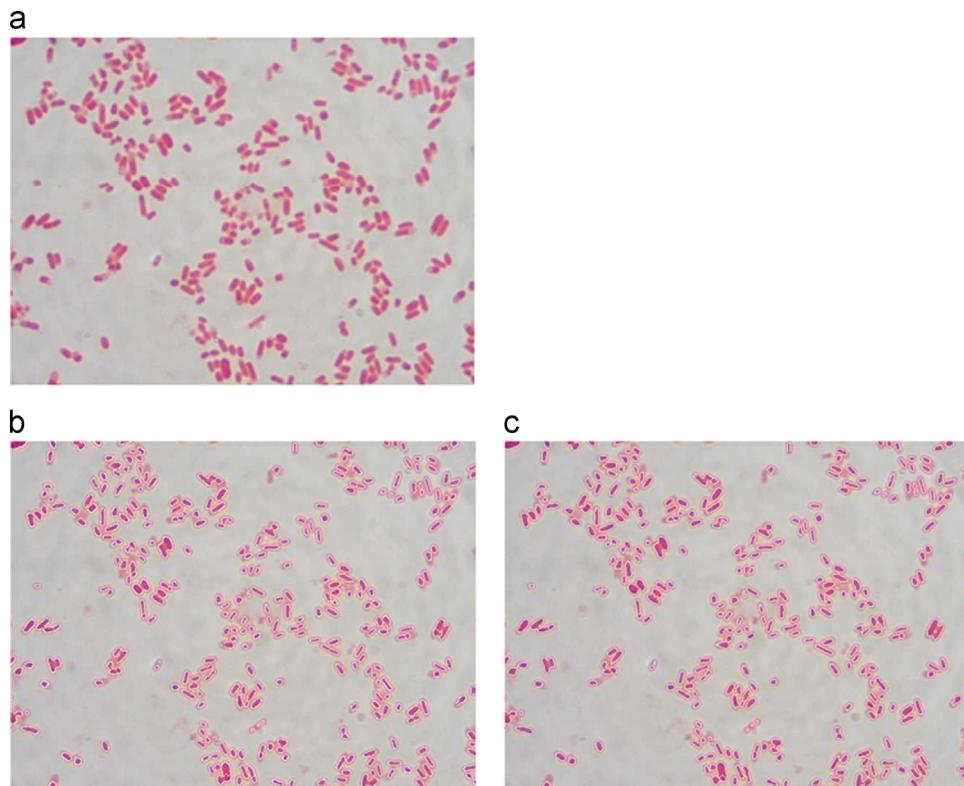


Fig. 14. The comparison between gradient-weighted distance transform watershed and gradient-barrier watershed (our method) on the *E. coli* image. (a) Original *E. coli* image. (b) Segmentation result of gradient-weighted distance transform watershed. (c) Segmentation result of gradient-barrier watershed, the gradient threshold is 10.

Table 2

The comparison between gradient-weighted distance transform watershed and gradient-barrier watershed (our method) on the *E. coli* image shown in Fig. 14.

	Ground truth	Correctly segmented	Merged	Split	Missed	Incorrect boundary
Gradient-weighted distance transform	333	283	22	16	4	8
Gradient-barrier watershed	333	294	22	9	4	4

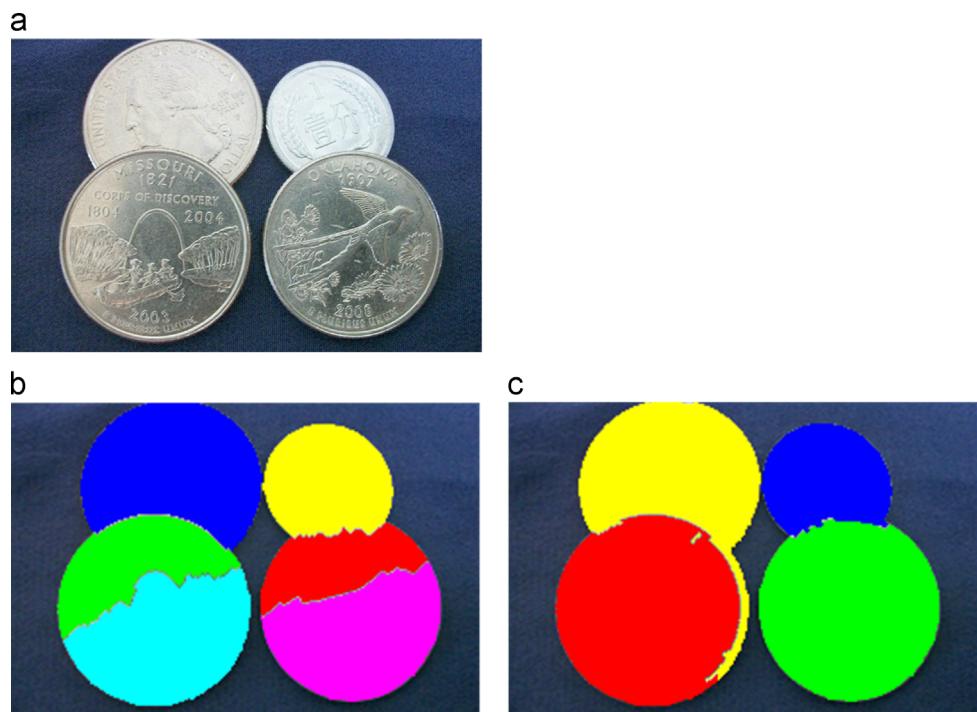


Fig. 15. The comparison between gradient-weighted distance transform watershed and gradient-barrier watershed (our method) on the coin image. (a) Original coin image. (b) Segmentation result of gradient-weighted distance transform watershed. (c) Segmentation result of gradient-barrier watershed, the gradient threshold is 20.

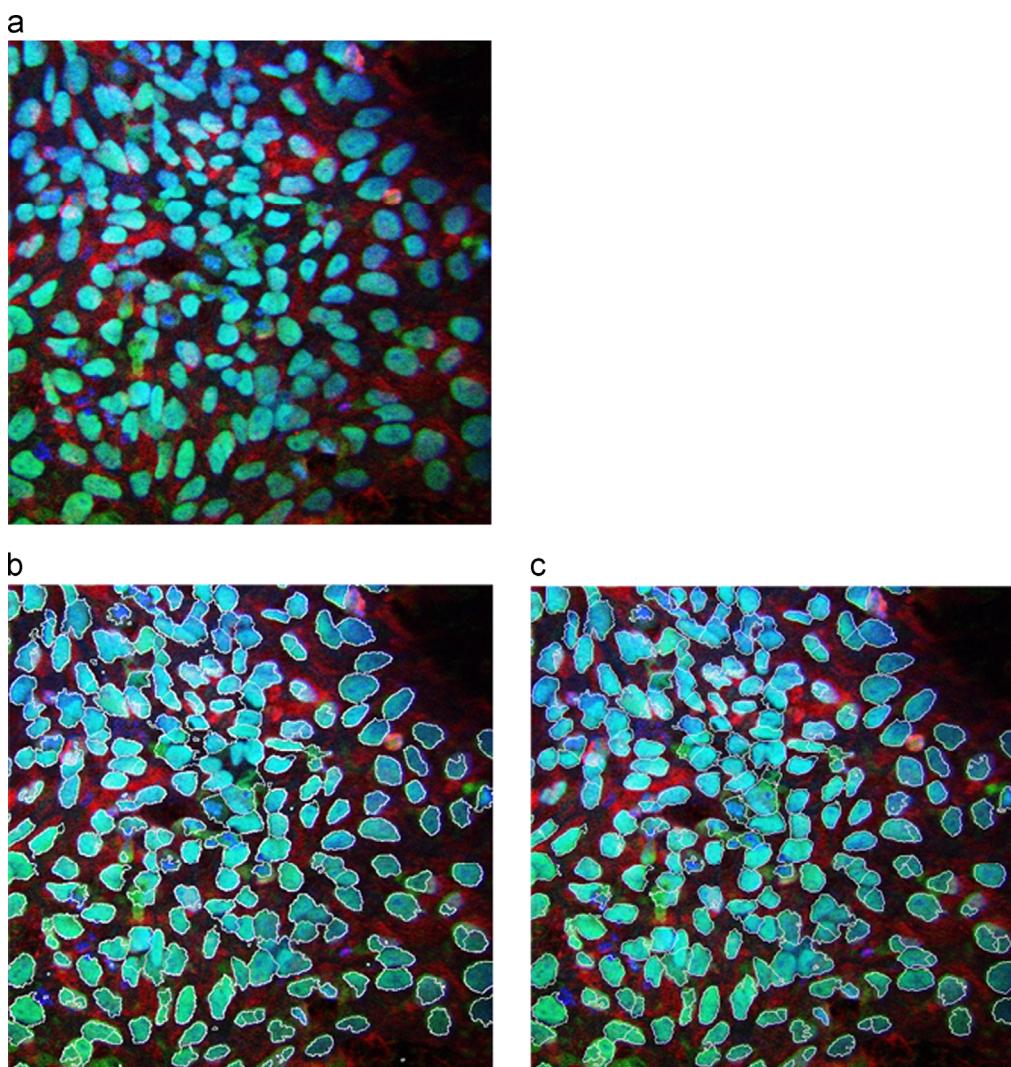


Fig. 16. The segmentation result for cell nuclei image. (a) Original cell nuclei image. (b) Segmentation result of gradient-weighted distance transform watershed. (c) Segmentation result of gradient-barrier watershed, the gradient threshold is 30.

Table 3

The segmentation result for cell nuclei image.

	Ground truth	Correctly segmented	Merged	Split	Missed	Incorrect boundary
Gradient-weighted distance transform	211	154	27	22	0	8
Gradient-barrier watershed	211	158	18	22	2	11

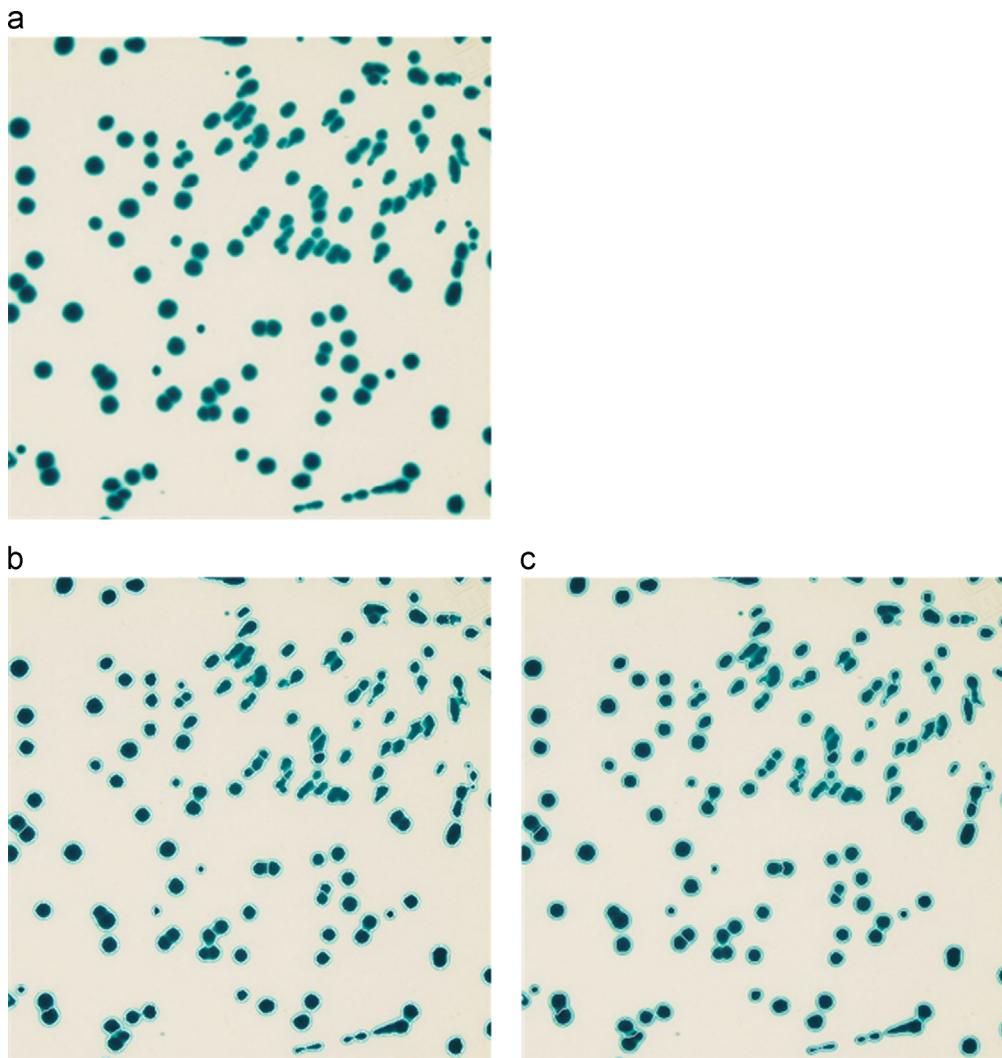


Fig. 17. The segmentation result for another *E. coli* image. (a) Original *E. coli* image. (b) Segmentation result of gradient-weighted distance transform watershed. (c) Segmentation result of gradient-barrier watershed, the gradient threshold is 30.

Table 4The segmentation result for *E. coli* image shown in Fig. 17.

	Ground truth	Correctly segmented	Merged	Split	Missed	Incorrect boundary
Gradient-weighted distance transform	175	130	44	0	1	0
Gradient-barrier watershed	175	131	43	0	1	0

detected in one object, and typically they are fairly close to each other. We then apply a clustering step on all the markers to group the multiple markers within the same object into one. To better incorporate the intensity information into the geometric information for conducting the watershed, we propose the gradient-barrier watershed, in which the gradient in the overlapping region is used directly as the barrier to the water flow. The experiments carried out on various granular objects from distinct application backgrounds justify our approach.

Conflict of interest statement

None declared.

Acknowledgments

The support of the Office of Naval Research under grant N00014-06-1-0101 is gratefully acknowledged. The authors would

like to give special thanks to Prof. Surangi Punyasena and Stefan Johnsrud at Department of Plant Biology, University of Illinois, Urbana-Champaign. It is in the cooperation with them that the author first time to focus on the segmentation of granular objects such as pollen grains and cells [33]. Also some testing images on pollen and cells are provided by Prof. Surangi Punyasena's lab. The author would also like to thank Yanxiang Shi at Department of Chemistry, University of Illinois, Urbana-Champaign, to propose the idea for applying the algorithm on counting the colonies grown on LB agar plate and providing the images of colonies. Some of the testing images are provided by above people or obtained online as follows:

Fig. 8: The image is provided by Prof. Surangi Punyasena at Department of Plant Biology, University of Illinois, Urbana-Champaign.

Fig. 10: The image is from University of California, Irvine website: <http://jeeves.mmg.uci.edu/immunology/Assays/ANA.htm>.

Fig. 11: The image is provided by Yanxiang Shi at Department of Chemistry, University of Illinois, Urbana-Champaign.

Fig. 12: The image is from Nanotechnology Now website: http://nanotech-now.com/news.cgi?story_id=29185.

Fig. 13: The image is from University of Wisconsin, Oshkosh website: http://uwosh.edu/med_tech/teaching/ElementaryHe meWeb/LEARN%20ABOUT%20RBCS%20AND%20PLTS.htm.

Fig. 14: The image is from University of Cincinnati, Clermont College website: http://www.biology.clc.uc.edu/fankhauser/labs/microbiology/gram_stain/Gram_stain_images/index_gram_stain_images.html.

Fig. 15: The image is from the website: http://sjbbs.zol.com.cn/2/992_16253.html.

Fig. 16: The image is from the website: <http://newswise.com/images/uploads/2012/09/18/Ruiz-StemCellPHOTO2.jpg>.

Fig. 17: The image is from the website: <http://ecx.images-amazon.com/images/I/61c7UkwAeDL.jpg>.

References

- [1] S. Shetty, Isotropy criteria and algorithms for data clustering, Ph.D. dissertation in electrical and computer engineering, University of Illinois, Urbana-Champaign, 2011.
- [2] J.A. Hartigan, *Clustering Algorithms*, Wiley, NY, USA, 1975.
- [3] P. Bajcsy, N. Ahuja, Location- and density-based hierarchical clustering using similarity analysis, *IEEE Trans. Patt. Anal. Mach. Intell.* 20 (1998) 1011–1015.
- [4] D. Jiang, J. Pei, A. Zhang, DHC: A density-based hierarchical clustering method for time series gene expression data. In: Proceedings of Third IEEE Symposium on Bioinformatics and Bioengineering, 2003, pp. 393–400.
- [5] D. Zhou, Z. Cheng, C. Wang, H. Zhou, W. Wang, B. Shi, SUDEPHIC: Self-tuning density-based partitioning and hierarchical clustering, database systems for advanced applications, ser. in: Y. Lee, J. Li, K.-Y. Whang, D. Lee (Eds.), *Lecture Notes in Computer Science*, vol. 2973, Springer, Berlin/Heidelberg, 2004, pp. 69–108.
- [6] J.Z. Chiwoo Park, J.X. Huang, Yu Ding Ji, Segmentation, Inference and classification of partially overlapping nanoparticles, *IEEE Trans. Pattern Anal. and Mach. Intell.* 35 (3) (2013) 1, <http://dx.doi.org/10.1109/TPAMI.2012.163>.
- [7] S. Beucher, The watershed transformation applied to image segmentation, *Scanning Microsc.* 6 (Suppl.) (1992) 299–314.
- [8] N. Malpica, C. de Sol'orzano, J. Vaquero, A. Santos, I. Vallcorba, J. García-Sagredo, F. del Pozo, Applying watershed algorithms to the segmentation of clustered nuclei, *Cytometry Part A* 28 (4) (1997) 289–297.
- [9] G. Lin, U. Adiga, K. Olson, J.F. Guzowski, C.A. Barnes, B. Roysam, A hybrid 3D watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks, *Cytometry Part A* 56A (2003) 23–36.
- [10] F. Long, H. Peng, E. Myers, Automatic segmentation of nuclei in 3D microscopy images of *C. elegans*. In: Proceedings of the 4th IEEE International Symposium on Biomedical Imaging: from Nano to Macro, ISBI, 2007, pp. 536–539.
- [11] P.S. Umesh Adiga, B.B. Chaudhuri, An efficient method based on watershed and rule-based merging for segmentation of 3-D histo-pathological images, *Patt. Recognit.* 34 (2001) 1449–1458.
- [12] F. Tek, A. Dempster, I. Kale, Blood cell segmentation using minimum area watershed and circle radon transformations, *Mathematical Morphology: 40 Years On, Computational Imaging and Vision* 30 (2005) 441–454.
- [13] J. Cheng, J. Rajapakse, Segmentation of clustered nuclei with shape markers and marking function, *IEEE Trans. Biomed. Eng.* 56 (2009) 741–748.
- [14] G. Lin, M. Chawla, K. Olson, J. Guzowski, C. Barnes, B. Roysam, Hierarchical, model-based merging of multiple fragments for improved three-dimensional segmentation of nuclei, *Cytometry Part A* 63 (1) (2005) 20–33.
- [15] N. Salman, Image segmentation based on watershed and edge detection techniques, *Int. Arab J. Inf. Technol.* 3 (2) (2006).
- [16] P. Bamford, B. Lovell, Unsupervised cell nucleus segmentation with active contours, *Signal Process.* 71 (2) (1998) 203–213.
- [17] S. Nath, K. Palaniappan, F. Bunyak, Cell segmentation using coupled level sets and graph-vertex coloring. In: *Proceedings of the 9th International Conference on Medical Image Computing and Computer-Assisted Intervention MICCAI'06*, Copenhagen, Denmark, Springer, 2006, pp. 101–108.
- [18] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, *Int. J. Comput. Vis.* 1 (4) (1988) 321–331.
- [19] T. Chan, L. Vese, Active contours without edges, *IEEE Trans. Image Process.* 10 (2) (2001) 266–277.
- [20] L. Vese, T. Chan, A multiphase level set framework for image segmentation using the Mumford and Shah model, *Int. J. Comput. Vis.* 50 (3) (2002) 271–293.
- [21] Y. Chen, H. Tagare, S. Thiruvenkadam, F. Huang, D. Wilson, K. Gopinath, R. Briggs, E. Geiser, Using prior shapes in geometric active contours in a variational framework, *Int. J. Comput. Vis.* 50 (3) (2002) 315–328.
- [22] A. Foulonneau, P. Charbonnier, F. Heitz, Multi-reference shape priors for active contours, *Int. J. Comput. Vis.* 81 (1) (2009) 68–81.
- [23] F. Lecumberry, A. Pardo, G. Sapiro, Simultaneous object classification and segmentation with high-order multiple shape models, *IEEE Trans. Image Process.* 19 (3) (2010) 625–635.
- [24] Y. Zhang, B. Matuszewski, Multiphase active contour segmentation constrained by evolving medial axes. In: *Proceedings of the 16th IEEE International Conference on Image Processing*, Cairo, Egypt, 2009, pp. 2993–2996.
- [25] M. Marcuzzo, P. Quelhas, A. Mendonca, A. Campilho, Evaluation of symmetry enhanced sliding band filter for plant cell nuclei detection in low contrast noisy fluorescent images, *Image Anal. Recognit.* (2009) 824–831.
- [26] P. Quelhas, M. Marcuzzo, A. Mendonca, A. Campilho, Cell nuclei and cytoplasm joint segmentation using the sliding band filter, *IEEE Trans. Med. Imaging* 29 (8) (2010) 1463–1473.
- [27] S. Beucher, Watershed: hierarchical segmentation and waterfall algorithm, in: J. Serra, P. Soille (Eds.), *Mathematical Morphology and Its Applications to Image Processing*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994, pp. 69–76.
- [28] L. Najman, M. Schmitt, Geodesic saliency of watershed contours and hierarchical segmentation, *IEEE Trans. Patt. Anal. Mach. Intell.* 18 (12) (1996) 1163–1173.
- [29] K.Z. Mao, P. Zhao, P.H. Tan, Supervised learning-based cell image segmentation for P53 immunohistochemistry, *IEEE Trans. Viomed. Eng.* 53 (2006) 1153–1163.
- [30] F. Meyer, Levelings, image simplification filters for segmentation, *J. Math. Image Vis.* 20 (2004) 59–72.
- [31] C. Wählby, I.-M. Sintorn, F. Erlandsson, G. Borgefors, E. Bengtsson, Combining intensity, edge, and shape information for 2D and 3D segmentation of cell nuclei in tissue sections, *J. Microsc.* 215 (2004) 67–76.
- [32] N.A. Johnsrud, Threshold selection method from gray-level histograms, *IEEE Trans. Syst. Man Cybern.* 9 (1979) 62–66.
- [33] S. Johnsrud, H. Yang, A. Nayak, S.W. Punyasena, Semi-automated segmentation of pollen grains in microscopic images: a tool for three imaging modes, *Grana* 52 (3) (2013) 181–191.
- [34] O. Ronneberger, W. Qing, H. Burkhardt, Fast and robust segmentation of spherical particles in volumetric data sets from brightfield microscopy. In: *Proceedings of the 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, ISBI 2008, 2008, pp. 372–375.
- [35] S.H. Landsmeer, E.A. Hendriks, L.A. de Weger, J.H.C. Reiber, B.C. Stoel, Detection of pollen grains in multifocal optical microscopy images of air samples, *Microsc. Res. Tech.* 72 (2009) 424–430.
- [36] H. Samet, M. Tamminen, Efficient component labeling of images of arbitrary dimension represented by linear bintrees, *IEEE Trans. Patt. Anal. Mach. Intell.* 10 (1988) 579.
- [37] L. He, Y. Chao, K. Suzuki, A run-based two-scan labeling algorithm, *IEEE Trans. Image Process.* 17 (5) (2008) 749–756.
- [38] M.B. Dillencourt, H. Samet, M. Tamminen, A general approach to connected-component labeling for arbitrary image representations, *J. ACM* 39 (2) (1992) 253–280.
- [39] N. Jamil, H.C. Soh, T.M.T. Sembok, Z.A. Bakar, A modified edge-based region growing segmentation of geometric objects, visual informatics: sustaining research and innovations, *Lect. Notes Comput. Sci.* 7066 (2011) 99–112.
- [40] A. Rosenfeld, J.L. Pfaltz, Sequential operations in digital picture processing, *J. Assoc. Comput. Mach.* 13 (4) (1966) 471–494.
- [41] P.F. Felzenszwalb, D.P. Huttenlocher, Distance transforms of sampled functions, *Theor. Comput.* 8 (2012) 415–428.

Huiguang Yang is a Ph.D. student in Electrical and Computer Engineering, University of Illinois, Urbana-Champaign. He received his B.S. in Environmental Engineering, Tsinghua University, Beijing, China, and received his M.S. in Atmospheric Sciences, University of Illinois, Urbana-Champaign. His research in Electrical and Computer Engineering focuses on image processing and computer vision, such as image segmentation and object recognition.

Narendra Ahuja received his Ph.D. from the University of Maryland, College Park, in 1979. He is Donald Biggar Willet Professor in Department of Electrical and Computer Engineering, University of Illinois, Urbana-Champaign. His fields of professional interest are next generation cameras, 3D computer vision, video analysis, image analysis, pattern recognition, human computer interaction, image processing, image synthesis, and robotics.