8

Rock Particle Image Segmentation and Systems

Weixing Wang
Collage of Computer Science and Technology, Hubei University of Technology,
Wuhan, Hubei,
China

1. Introduction

As known, most important, and the hard part of pattern recognition for rock particles, is image segmentation. Segmentation can be divided into two steps, one is segmentation based on gray levels (called image binarization, sometimes) in which a gray level image is processed and converted into a binary image. Another is segmentation based on rock particle shapes in a binary image, in which overlapping and touching particles will be split, and over-segmented particles will be merged based on some prior knowledge such as shape and size etc.

Segmentation algorithms for monochrome (gray level) images generally are based on one of two basic properties of gray-level values: similarity and discontinuity. The principal approaches in the first category are based on thresholding, region growing, and region splitting and merging. In the second category, the approach is to partition an image based on abrupt changes in gray level. The principal areas of interest within this category are detection of isolated points and detection of lines and edges in an image.

The choice of segmentation of rock particle images based on similarity or discontinuity of the gray-level values depends on both developed sub-algorithms and applications.

Rock particle images have their own characteristics compared to other particle images. Generally speaking, under the frontlighting illumination condition which is common case, rock particle images have the characteristics: (1) uneven background and foreground for which a simple thresholding algorithm cannot be applied to segment the images; (2) each rock particle may possess a textured surface and multiple faces, which often causes an over-segmentation problem; (3) rock particle overlapping each other, which hides parts of a particle, or causes breaks of the boundaries of rock particles; (4) touching rock particles forming a large cluster; (5) rain, snow, or much fine material making rock particle images clump together.

Rock particles may be densely packed or be separated mostly on a background. The former case is more difficult to process than the latter. As well known, most systems for rock particle images were developed based on simple thresholding algorithms (some of them combined with morphological segmentation algorithm) and boundary detection algorithms. The segmentation algorithm designing is application (here, the type of rock particle images) dependent. In this chapter, the author summarize own segmentation approaches for rock particle images, they are: (1) an algorithm based on edge detection; (2) an algorithm based on region split-and-merge; (3) an adaptive thresholding algorithm; and (4) an algorithm for splitting touching particles in a binary image.

Source: Pattern Recognition Techniques, Technology and Applications, Book edited by: Peng-Yeng Yin,
ISBN 978-953-7619-24-4, pp. 626, November 2008, I-Tech, Vienna, Austria

The size, shape and texture of rock particles are very important characteristics of the physical properties for the geology research and rock particle production industry and mining industry. In mining, the size and shape distributions of fragments affect not only rock blasting, but also the whole mining production sequence. In the quarry manufacture, the size, shape and texture of rock particles must fit the requirements of customers, such as high way and rail way construction companies, the different companies in the building industries, etc. In geology, the size, shape and texture of gravel and sedimentary deposits are often used for analyzing and describing local geological properties in a certain region. Hence, rock particle size, shape and texture are widely applied and studied in both industries and research organizations.

Mechanical and manual measurement methods are traditional methods. Mechanical methods such as screens and classifiers will often separate rock particles according to their shapes, as well as density and size. In the laboratory, sieving of dry material is possible at sizes as fine as 0.05 mm and classification would not be applied to sizes greater than 0.1 mm. In industrial practices, it is impractical for screening the material sized below 3 mm, when the moisture is up to 3-10% by weight. These devices physically separate particles and often characterize the distribution of size in a limited size of a particle sample. The form of rock particle is often manually measured in a laboratory, in this way, three or more parameters such as the rock particle length, width, and thickness are often measured respectively for every individual particle. However, sieving and manual measurement methods are manpower and time consuming methods, in which the size of a sample is limited, and the measuring results are only suitable for the simple shape such as cube and cuboid. The results measured by different persons will be different in manual measurement [1]. In order to increase accuracy and speed of measurement, reduce manpower consuming, and enlarge sample size, new measurement methods are needed to be developed based on currently developed techniques. The image analysis method is one of these relatively new methods in engineering geology. For developing this kind of measurement methods or systems, the knowledge of geology or mining engineering, the techniques of image analysis and computer vision, as well as the skills of computer software development are needed.

As computers are widely used today, the cost of an image system is often relatively low, and rock particle size, shape and texture analysis can thereby be handled easily and quickly. Image analysis is a subset of the wider field computer vision, which aims at imitating biological or human vision performance. Identifying and separating overlapping objects from each other, is something that human vision can do with surprising (uncanny) ease. It is still an open question how to achieve this algorithmically in computer vision..

2. Literature review

The earliest image analysis system for rock particles was developed by Gallagher [2] in 1976. In his PHD study, he set up a system aimed to measure fragment size parameters on a conveyor belt. The camera was mounted above the particle stream with its optical axis aligned normal to the moving bed of particles. The size distribution of the fragments was then computed by finding the spacing of edges with a chord sizing method. Nyberg (1982) [3] presented an image system scanning chord size on an edge image of fragments in a rockpile. During the past fifteen years, image analysis for rock particles has become a hot topic of research, and a number of image systems have been developed for measuring rock particles in different applications such as gravitational flows, conveyor belts, rockpiles, and laboratories, and some of them are under development. The researches and developments have been and are carried out in many countries such as Sweden, Germany, France, England, Australia, Canada, United States, China, South Africa, and Spain etc.[4-11].

Generally speaking, a system consists of three parts: image acquisition, image segmentation and particle size and shape measurement. Image acquisition is about how to set up an image acquisition system to acquire rock particle images of good quality under different work conditions, which is often strongly application dependent. Image segmentation is an important part in the whole system. The design of an image segmentation algorithm depends on the characteristics of rock particle images. The image segmentation results affect the accuracy of measurement of size, shape and texture of rock particle particles. The basic requirement for size, shape and texture measurement is that the measurement results should be reproducible, and should reflect as much information as possible.

According to algorithms or methods of image segmentation, the existing systems could be classified into at least four classes. They are: (1) thresholding on histogram of gray levels [12-18]; (2) boundary tracing or edge detection [2-3, 7-8, 19-33]; (3) region growing or merge & split [9-10, 34-37]; and (4) thresholding and granulometry (= morphology segmentation on a binary image) [38-41].

Systems based on thresholding of histogram of gray levels were applied in some applications in which rock particle images have a uniform background, and rock particle possessing less surface texture. The typical application is measuring rock particles in a gravitational flow. Recently, such systems are used both in laboratories and for conveyor belts in the field [15-18]. The system uses backlighting illumination (a special lighting condition is constructed) to acquire rock particle images from a free falling rock particle stream, the acquired images being almost binarised ones. Therefore, a simple threshold can separate rock particles and background easily.

There is a number of systems developed based on boundary tracing or edge detection algorithms. The early systems mentioned before [2-3], used a difference operator to obtain a gradient magnitude image from a gray level image, then binarised the gradient magnitude image. The binarised image is the image with contours of particles. In most cases, the segmentation results are not satisfactory due to the fact that the contours of particles are not closed curves, and false edges exist. In order to overcome the problems, some recent systems include procedures (sub-algorithms) for gap linking, false edge elimination and curve closing. Some typical examples are summarized below.

In Lin, Yen and Miller's system (1995) [31], an image of overlapped rock fragments, taken from a moving conveyor belt, is first smoothed with an edge preserving filter, secondly detected by an edge detection operator (Canny's algorithm), then processed by edge linking and edge gap filling, finally followed by segment connection. Transforming the intensity function of the processed image for the desired intensity regions smooths the original image. The Canny edge detector is used with a so-called hysteresis thresholding algorithm to extract edges from the smoothed image. Supplementary to edge detection, edge linking and gap filling functions are added in the algorithm.

Kemeny et al. [24-27] system has been used in many cases. The system enhances an image by equalizing histogram of gray levels, then thresholds the enhanced image to separate void spaces among particles (non-particle regions), so-called shadows, from particles. Meanwhile a gradient magnitude image is obtained by using a Sobel difference operator. Particles are delineated by searching for large gradient paths ahead of sharp convexities of the shadows to separate clusters of touching rock particles. Using a morphological segmentation algorithm splits the remaining touching particles.

Norbert, Tom and Franklin [21-22 setup a system since 1988]: The work sequence of the segmentation algorithm is similar to Kemeny's one. It includes two steps. The first step is to segment particles by use of several conventional image processing techniques, including the use of thresholding and gradient operators. In this step, the faint shadows between adjacent particles are detected, and the work step is available for clean images with lightly textured rock surfaces. The second step uses a number of reconstruction techniques to further delineate particles which are only partly outlined during the first step. In the second step, the algorithm is just for closing particle contours. Bedair 1996 in his Ph.D. study, developed a similar particle contour closing algorithm, the detail description can be found in [32-33].

In morphological segmentation of rock particle images, the thresholded binary image is the objective. Two kinds of algorithms have been used in the image analysis systems for rock particles: one is granulometry, the other is Watershed algorithm.

The general idea of the Granulometry algorithm is that in order to simulate the sieving analysis, one can generate a series of squares (a maximum square inside of an particle) to obtain size distribution. In this algorithm, the complex segmentation is avoided. The ideal case is that particles have some regular shape (e.g. circle, square, or diamond). It is mainly based on the functions of opening and closing with a certain structure element, and distance transformation. The systems applying the algorithm can be found in [39-41].

Yen, Lin and Miller's method (1994) [38] - a derivative of the Watershed segmentation: It includes seven steps: (1) "Edge Preserving Smoothing" technique is applied to the overlapped fragment image to eliminate the fluctuation of highs and lows on the particle surface as much as possible but preserve the edge points; (2) the Sobel edge detector is used to find the edges on the smoothed particle image; (3) A median filter is utilized to eliminate the noise on the edge image; (4) the smoothed edge image is subtracted from the original image to construct an "edge cutted" (EC) image; (5) a gray scale morphology erosion is applied to shrink EC particle image to an extent such that no overlap occurs; (6) the "Otsu" thresholding algorithm [74] is used to shrunk, non-overlapped image and a labeling procedure used to identify each mark; (7) Once these inside markers have been located the "Odered Queue Watershed" algorithm can then be applied to the original particle image to separate the overlapped particles. The segmentation result is not very satisfactory even to well sorted particles with a good background.

The segmentation algorithms of region growing or merge & split for rock particle images [9-10, 34-37] were and are mainly developed by the author. The chapter will discuss that segmentation algorithm. All the four kinds of segmentation algorithms have been developed more or less in the study. The different developed algorithms have been chosen for different applications. The developed algorithms are described and compared too.

3. Image segmentation algorithm for rock particles

As mentioned before, most important, and the hard part of computer vision for rock particles, is segmentation. Segmentation can be divided into two steps, one is segmentation based on gray levels (called image binarization, sometimes) in which a gray level image is processed and converted into a binary image. Another is segmentation based on particle shapes in a binary image, in which overlapping and touching particles will be split, and oversegmented particles will be merged based on some prior knowledge such as shape and size etc.

3.1 The algorithms based on edge detection

3.1.1 Crucial algorithm edge detection

As most developed segmentation algorithms in the existing systems for rock particle images, an algorithm based on edge detection was also developed in this study. The main parts of the algorithm are (1) image smoothing; (2) edge detection by an edge operator; (3) thresholding on the obtained gradient magnitude image; and (4) noise edge deleting and edge gap linking.

After image smoothing (e.g. Gaussian smoothing), the Canny edge detector [59] is applied on the smoothed image. The gradient magnitude image obtained by Canny's operator, is thresholded by the P-tile thresholding algorithm, the value of the P-tile is chosen according to characteristics of rock particle images. Before noise edges deleting and edge gap linking, the thresholded image is thinned by a thinning function. After this operation, all the end points of edges (lines or curves) are detected, and small gaps between two edges are linked, where some end points disappear. All the edges of the end points are eroded from the end points within a certain length L_E(a number of pixels), the short edges of length $< L_E$, are removed. The remaining edges are then dilated to recover their original states. Finally, the gaps (e.g. the length of a gap is less than 20 pixels) between edges are linked.

As examples, two densely packed rock particle images are segmented by the algorithm (see Fig. 1). In the present stage, the algorithm can not provide closed curve for each individual particle, but the segmentation result can be used for estimation of average size of densely packed rock particle particles.

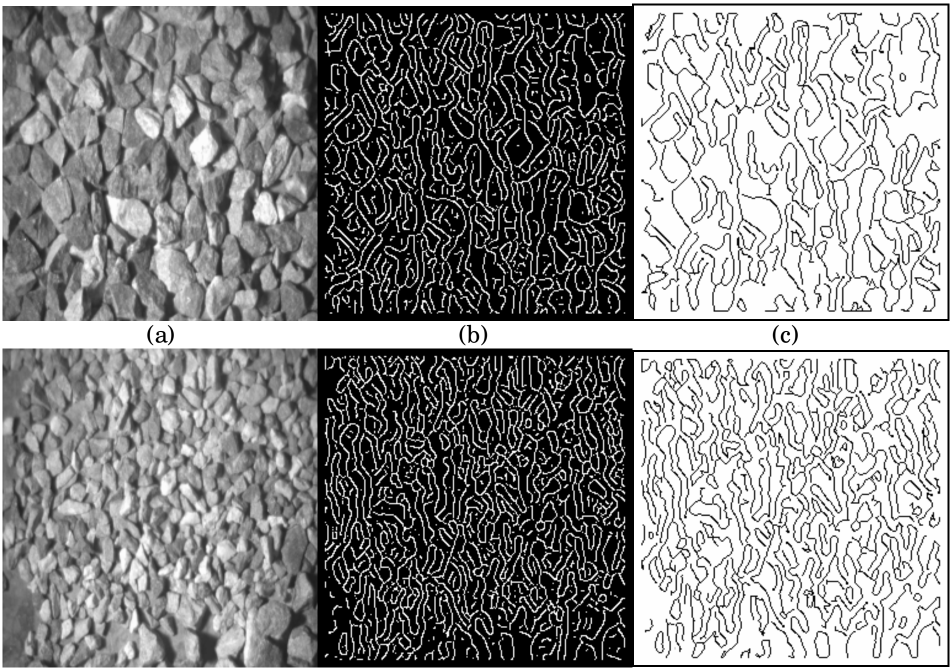

(a) (b) (c)
(d) (e) (f)

Fig. 1. Segmentation based on edge detection. (a) Original image #1. (b) Image after Canny operation on (a). (c) Image after deleting noise edges and gap linking on (b). (d) Original image #2. (e) Image after Canny operation on (d). (f) Image after noise edges deleting and gap linking on (e).

3.1.2 The algorithm of one-pass boundary detection

The goal of edge detection in our case is to quickly and clearly detect the boundaries of particles, it is not necessary to close every particle's boundary (it is too hard), but it should produce less gaps on boundaries and less noise edges on the particles. To reach this goal, we tested several widely used edge detection algorithms for a typical particle image; in Fig. 2, (a) original image (150x240x8 bits), (b) Sobel edge detection result that includes too much white noise, (c) Robert edge detection result that is mass, (d) Laplacian edge detection result that miss boundaries much, (e) Prewitt edge detection result that is similar to (a), (f) and (g) Canny edge detection results which are thresholding value dependent, and (h) the result from our developed one-pass boundary detection algorithm. By comparing results from the seven tests, the new algorithm gives the best edge (boundary) detection result. Our algorithm [53] is actually a kind of ridge detector (or line detector).

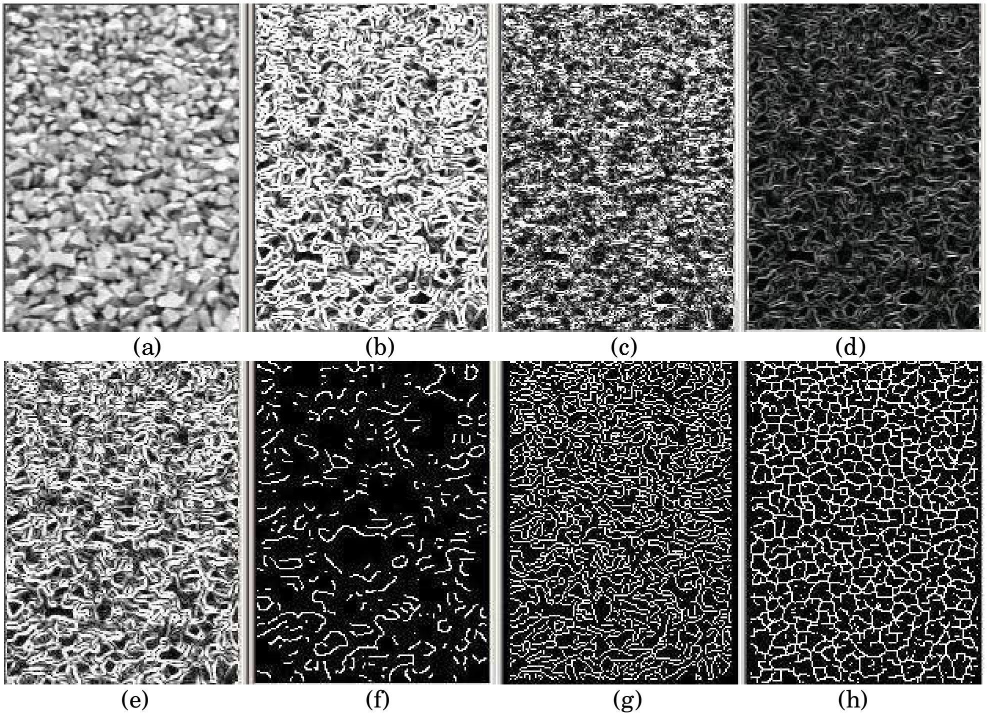


Fig. 2. Testing of edge detection algorithms: (a) Original image; (b) Sobel detection; (c) Robert detection; (d) Laplacian detection; (e) Prewitt detection; (f) Canny detection with a high threshold; (g) Canny detection with a low threshold; and (h) Boundary detection result by the new algorithm.

To overcome the disadvantages of the above first six edge detection algorithms, we studied a new boundary detection algorithm (Fig. 2 (h)) based on ridge (or valley) information. We use the word valley as an abbreviation of negative ridge. The algorithm is briefly described as follows:

A simple edge detector uses differences in two directions: $\Delta_x = f(x+1, y) - f(x, y)$ and $\Delta_y = f(x, y+1) - f(x, y)$, where $f(x, y)$ is a grey scale image.

In our valley detector, we use four directions. Obviously, in many situations, the horizontal and vertical grey value differences do not characterize a point, such as P (in Fig. 3), well.

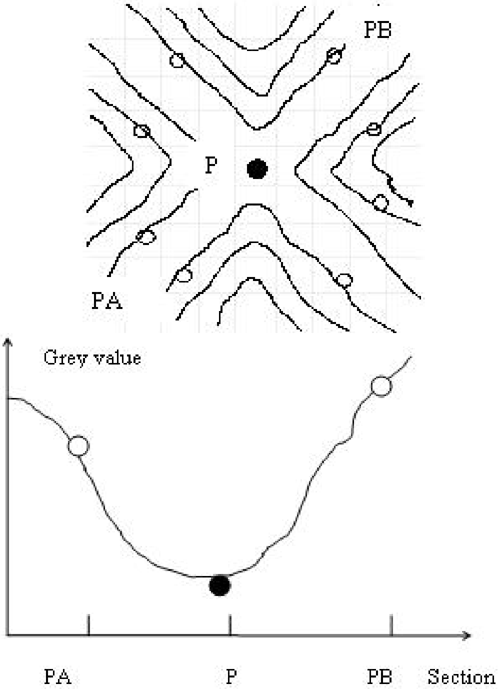


Fig. 3. Examine the point P, determining if it is a valley pixel, or not. Circles in the sparse (i, j)-grid. It moves for each P ∈ (x, y)-grid. (a) A grey value landscape over layered with a sample point grid. (b) PA-PB section.

In Fig. 3, we see that P is surrounded by strong negative and positive differences in the diagonal directions: $\nabla_{45} < 0$, and $\Delta_{45} > 0$, $\nabla_{135} < 0$, and $\Delta_{135} > 0$, whereas, $\nabla_0 \approx 0$, and $\Delta_0 \geq 0$, $\nabla_{90} \approx 0$, and $\Delta_{90} \approx 0$, where Δ are forward differences: $\Delta_{45} = f\left(i+1, j+1\right) - f\left(i, j\right)$, and ∇ are backward differences: $\nabla_{45} = f\left(i, j\right) - f\left(i-1, j-1\right)$ etc. for other directions. We use $\max\left(\Delta_\alpha - \nabla_\alpha\right)$ as a measure of the strength of an edge point. It should be noted that we use sampled grid coordinates, which are much more sparse than the pixel grid $0 \leq x \leq n$, $0 \leq y \leq m$. f is the original gray value image after slight smoothing.

What should be stressed about the valley edge detector is:

a. It uses four instead of two directions;

b. It studies value differences of well separated points: the sparse $i \pm 1$ corresponds to $x \pm L$ and $j \pm 1$ corresponds to $y \pm L$, where $L >> 1$, in our case, $3 \leq L \leq 7$. In applications, if there are closely packed particles of area > 400 pixels, images should be shrunk to be suitable for this choice of L. Section 3 deals with average size estimation, which can guide choice of L;

c. It is nonlinear: only the most valley-like directional response $\left(\Delta_\alpha - \nabla_\alpha\right)$ is used. By valley-like, we mean $\left(\Delta_\alpha - \nabla_\alpha\right)$ value. To manage valley detection in cases of broader valleys, there is a slight modification whereby weighted averages of $\left(\Delta_\alpha - \nabla_\alpha\right)$-expressions are used.
$w_1\Delta_\alpha\left(P_B\right)+w_2\Delta_\alpha\left(P_A\right)-w_2\nabla_\alpha\left(P_B\right)-w_1\nabla_\alpha\left(P_A\right)$, where, P_A and P_B are shown in Fig. 3.
For example, w_1=2 and w_2=3 are in our experiments.

d. It is one-pass edge detection algorithm; the detected image is a binary image, no need for further thresholding.

e. Since each edge point is detected through four different directions, hence in the local part, edge width is one pixel wide (if average particle area is greater than 200 pixels, a thinning operation follows boundary detection operation);

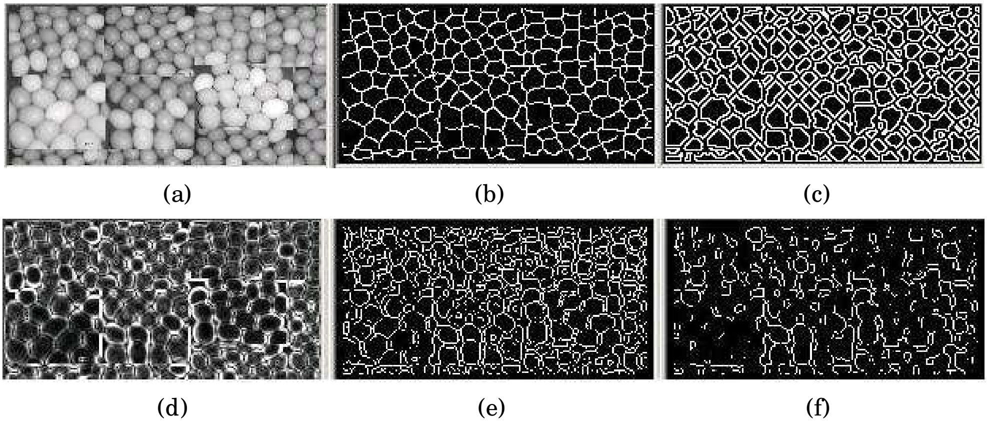
f. It is not sensitive to illumination variations, as shown in Fig. 4, an egg sequence image. On the image, illumination (or egg color) varies from place to place, for which, some traditional edge detectors (Sobel and Canny etc.) are sensitive, but the new edge detector can give a stable and clear edge detection result comparable to manual drawing result.

The algorithm has been tested for a number of images. It works satisfactory in several kinds of applications, and the testing results are shown in Figs. 5 -8.

In Fig. 5, the froth is very small, say, 43 pixels per bubble on average, it is hard to delineate by using a common image segmentation algorithm.

The image in Fig. 6 is different from the image in Fig. 5: the bubble size varies much; the white spots can clearly be seen on relative large bubbles. The ordinary edge detector may just detect the edges of the white spots, which are not the boundaries of the bubbles.

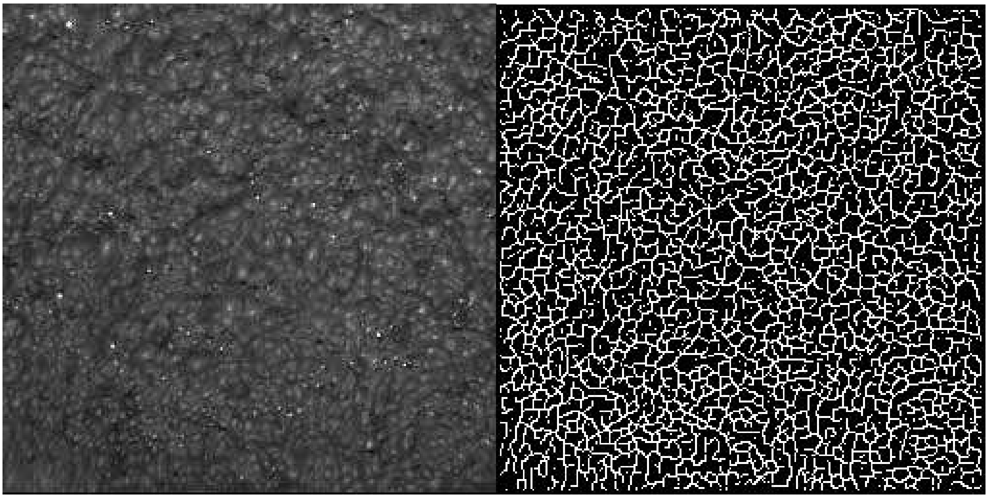
The image in Fig. 7 includes a mass of rough surface particles (average area is about 45 pixels), the new algorithm works well even for this kind of images.



(a) (b) (c)

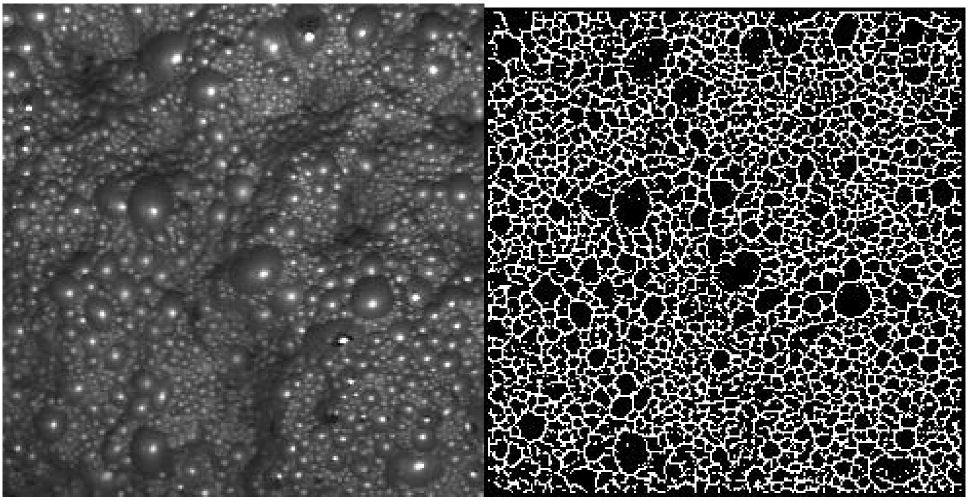
(d) (e) (f)

Fig. 4. Egg image test: (a) original image (400x200 pixels), (b) new algorithm result, (c) manual drawing result (180 eggs), (d) Sobel edge detection result, and (e) and (f) Canny edge detection results with different thresholds.



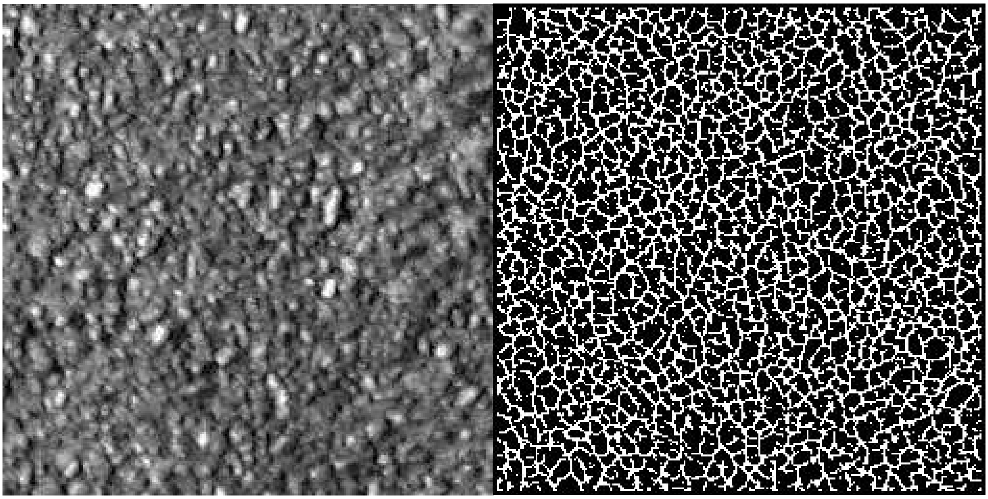
(a) (b)

Fig. 5. Froth image of well sorted bubbles (image size 256x256, about 1516 bubbles): (a) Original image, (b) Boundary detection result.



(a) (b)

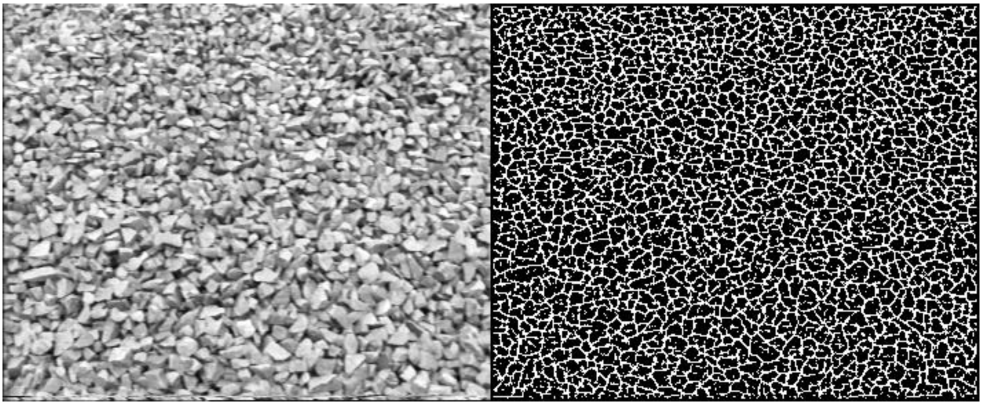
Fig. 6. Froth image of non-well sorted bubbles (image size 256x256, about 1421 bubbles): (a) Original image, (b) Boundary detection result.



(a) (b)

Fig. 7. Soil image of well sorted particles (image size 256x256, about 1445 particles): (a) Original image (particle surface is very rough), (b) Boundary detection result.

In Fig. 8, the image consists of a number of crushed aggregate particles, 47 pixels per aggregate particle on average. Even for the non-smooth (non-rounded) surface particles, the new edge detection algorithm can give a good detection result.



(a) (b)

Fig. 8. Crushed aggregate image of well sorted particles (image size 356x288, about 2173 particles): (a) Original image, (b) Boundary detection result.

The new algorithm includes only some kind of differentiation - one of the three operations (differentiation, smoothing and labeling) by comparing to ordinary edge detectors. It is a kind of line detection algorithm, but detecting lines in four directions.

After boundary detection, the edge density will be counted and converted to particle size, the next section presents our particle size estimation algorithm.

3.2 The algorithm based on split-and-merge

For the images of densely packed rock particle particles, the above segmentation algorithm can provide average size rather than size distribution of particles. To meet the requirement of obtaining a size distribution of rock particle particles, a segmentation algorithm based on region split-and-merge was developed. The algorithm consists of three parts: (1) Suk & Chung algorithm-Single-pass split-and-merge [60]; (2) merging small regions into their adjacent large regions; (3) background split and regions merge based on shape of rock particle particles.

For a rock particle image, the Suk & Chung algorithm [60] is first applied to segment the rock particle image into small regions. However, this segmentation based on gray values, yields a number of the small regions amounting to tens up to hundreds times the real number of particles in an image. In order to reduce the number of the small regions, a merge procedure was developed, in which the two steps are included: (1) Find the small regions R_s ($< T_3$, T_3 is a threshold value); (2) Among R_j ($j = 1, 2, \ldots$), all the neighboring regions of R_s, find R_m ($j = m$) for which the common edges between R_s and R_j is maximal, and then merge R_s and R_m.

Sometimes, the whole rock particle image is not fully occupied by the particles. Parts of the non-particles regions or void spaces tend to be dark. To eliminate regions belonging to dark background, one may let regions of average gray value below a pre-defined threshold be re-classified as background, so-called "background split". The use of a pre-defined threshold is partly enabled by the normalization pre-processing procedure.

When the background is split from the image (i.e. the image is converted into a binary image), over-segmentation problem still exists in the binarized image. To overcome this problem, a procedure for merging regions based on shape of rock particle particles was constructed. In the merge procedure, three basic merge criteria were considered for two neighboring regions (or objects), they are: (1) their common boundary length is relatively long; (2) the gray value difference between two objects is not too large; and (3) if two objects are merged, the two junction points should not be concave points.

The whole algorithm work sequence is illustrated in the following Figures. An original rock particle image (from pavement) is shown in Fig. 9(a). In the first processing step, the image is merged and split into many small regions, each of them has an uniform gray intensity (see Fig. 9(b)). After merging small regions into their adjacent large regions, the result image is shown in Fig. 9(c), where the over-segmentation problem still exists. To reduce this kind of problem, in the last step, the merge procedure on a binary image, is used, and the result is shown in Fig. 9(d).

When a rock particle image is complex as shown in Fig. 10(a), one extra pre-processing algorithm has to be used before the segmentation algorithm is applied. The image pre-processing algorithm was designed to: (i) strengthen the edges among the rock particles; (ii) delete or decrease bright spots noises (= noise or irrelevant detail) in a rock particle image; and (iii) remove slowly varying intensity variations in a rock particle image. (i) is crucial because the contrast between touching or overlapping particle edges may be quite low, causing great difficulties for image segmentation. (ii) is very important for removing spots noise (e.g. fine material) and smooth particles. (iii) is of interest, since by making the image more homogeneous, eliminating some slowly sloping regions, segmentation is expected to work more efficiently. One example is shown in Fig. 10, to illustrate the usefulness of the

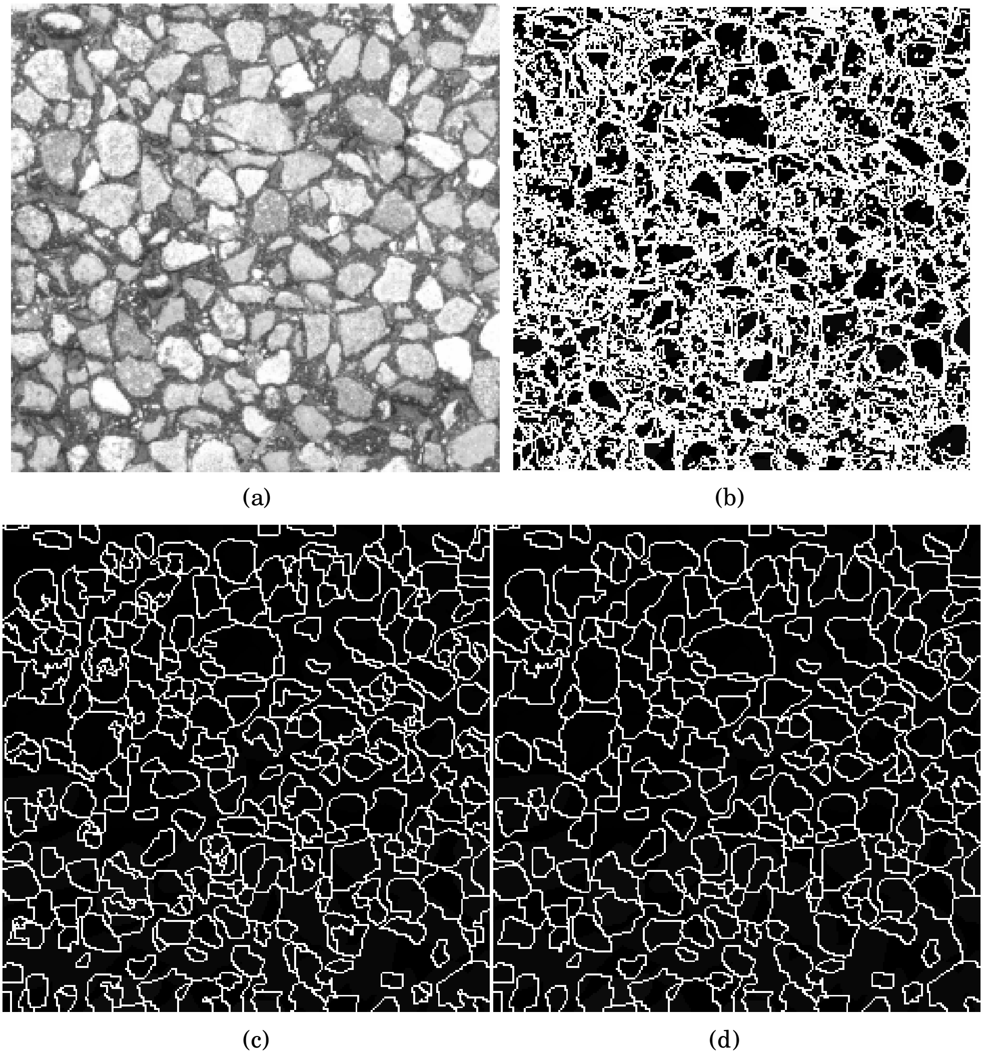
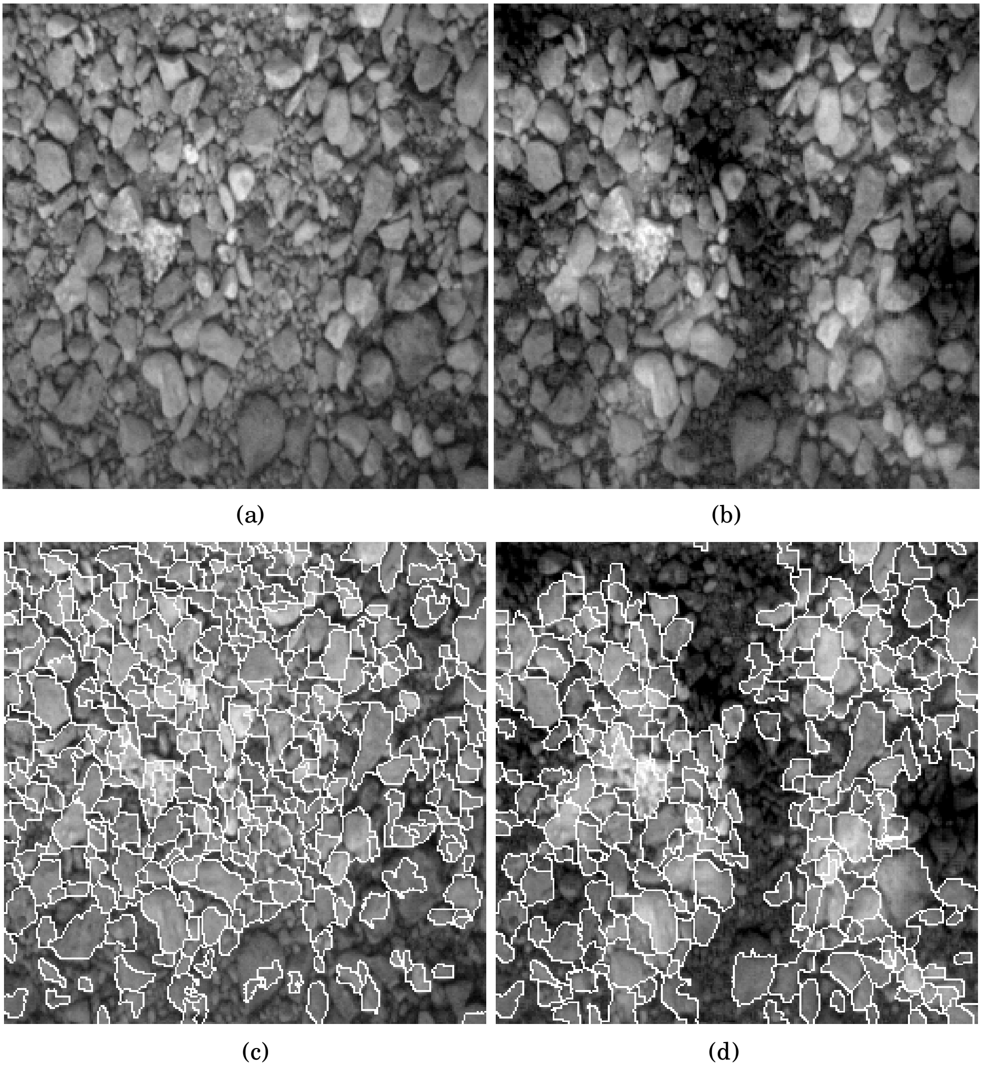

(a)
(b)
(c)
(d)

Fig. 9. Image segmentation based on split-and-merge. (a) Original image. (b) Result of split-and-merge. (c) Result after merging small regions into their adjacent large regions. (d) Image after regions merge based on shapes of rock particle particles.

image pre-processing algorithm. The segmentation result is not satisfactory because of without using the preprocessing procedure before segment the image.


(a) (b)

(c) (d)

Fig. 10. Comparison between two segmentation results by using the segmentation algorithm described in the following section. (a) Original image. (b) Image after the pre-processing. (c) Segmented result on the image in (a). (d) Segmented result on the image in (b).

Before the above procedure, a preprocessing step is operated. as the follows.
The principal objective of image pre-processing is to process an image so that the result is more suitable than the original image for a specific segmentation algorithm. For the ceramic surface image segmentation, in order to make images more suitable for segmentation, we use our pre-processing program to reduce image noises in three steps: (1) strengthen the

edges among the ceramic grains; (2) delete or decrease bright spots noises (= noise or irrelevant detail) in a ceramic image, and (3) remove slowly varying intensity variations in a ceramic image. Item (1) is crucial because the contrast between touching or overlapping grain edges may be quite low, causing great difficulties for image segmentation. Item (2) we have already discussed. Item (3) is of interest, since by making the image more homogeneous, eliminating some slowly sloping regions, segmentation is expected to work more efficiently.

We strengthen the edges by subtracting a gradient image (using Robert's difference operator) times a factor λ from fo, the original ceramic image, Eq. (1). We obtain a new image fn with more contrast along edges.

$$f_n(x,y) = f_o(x,y) - \lambda M(X,Y) \tag{1}$$

In Eq. (1), fn(x, y) is ceramic image after edge strengthening, fo(x, y) the original ceramic image, M(x, y) the magnitude image (based on fo) and λ a parameter, say $\lambda = 0.5$.

Next, a curved normalization surface T(x, y) is constructed, for which, a normalization value is assigned to each pixel, given by Eq. (2). In Eq. (2), μ0 and d0 are global mean grey value and standard deviation of fn, respectively, and μ and d are local mean grey value and standard deviation of fn (e.g. 16x16 window),

$$T = \mu - 0.2(d - d_0) - 0.5(\mu - \mu_0) \tag{2}$$

The image elements for which grey values are larger than T(x, y), (Here, T is used for grey value slicing, for finding bright regions. In Eq. (3), it is used for normalization.), will be processed through shrinking and expanding, so-called morphological operations, causing regions of width around 2 - 3 pixels, say narrow bright thin lines or bright spots, to vanish. In this case, the function T is used for detecting narrow or small bright regions.

$$f_N(x,y) = f_n(x,y) - T(x,y) + Const. \tag{3}$$

Narrow dark regions cannot be removed in this way since we then may destroy void space separating two grains. Slowly varying grey values can, locally, causes extra "shadows" in a grain, which makes segmentation difficult, for example, when separating away the background or when selecting threshold values for region merging and splitting, see the follows For that reason, the edge enhanced grey-level image fn is normalized by subtracting T, see Eq. (3), yielding fN.

By applying the procedures, the complicated images can be processed satisfactorily; the following figures show the results (Fig. 11).

3.3 The algorithms based on thresholding

When an rock particle image has a uniform background, or particles' gray intensity differs to their surrounding regions (local background regions), a segmentation algorithm based on thresholding is applicable. There is a number of thresholding algorithms published in literature [55]. They can be classified into global and local (adaptive) ones. In order to evaluate the existing global thresholding algorithms with respect to rock particle images, a comparison study has been carried out [55]. The comparison results show that for a rock particle image with a uniform background or a global background which can be distinguished from rock particles by human vision, the algorithms Optimal threshold (OPT), Between class variance (BCV), are the best choices for performing global threshold. One example is shown in Fig. 12. The original image was taken from a laboratory, and comprised of sand particles (of sizes 1 - 4 mm), the background gray value range is 130 - 250, and the range of gray values of particles is 10 -120. The BCV thresholding is presented in Fig. 12(b). The split algorithm summarized in next section split the touching particles.

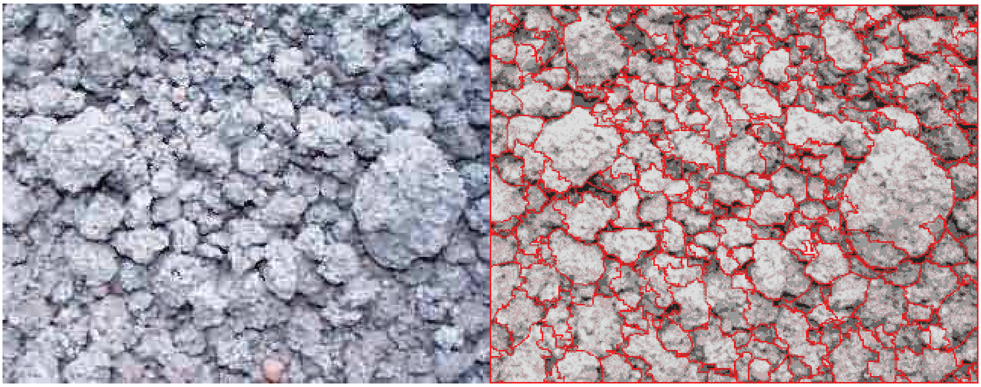
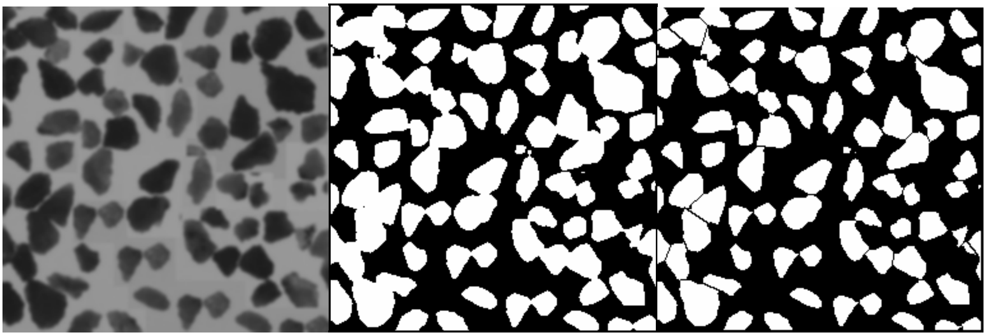


Fig. 11. Image without clear edges, particle surface is rough, non-uniform background



(a) (b) (c)

Fig. 12. Segmentation based on BCV global thresholding algorithm. (a) Original sand particle image. (b) Binarization result. (c) Touching particles were split by the split algorithm summarized in next section.

When the background gray level changes from part to part, and the gray level in some part of background is similar to that in some particle regions, the global thresholding cannot be used. In one case, when every particle region is surrounded by background, and the gray level of the local background is quite different to that of the surrounding particle region, an adaptive thresholding algorithm can be used. One special adaptive thresholding algorithm, a so-called recursive BCV algorithm, was developed for the rock particle images in this case. The developed algorithm assumes that the grey levels of *local* background are significantly higher than those of the particles (note that an object may consist of several particles touching each other to form a cluster); and that the range of the grey values in a particle is not too large. In practice the new algorithm processing sequence is: (1) the BCV algorithm is applied to the whole image for the initial thresholding round. Then, (2) for area A, a shape factor S and the range of grey levels Δf for *each* object is calculated. (3) For one object, if the area is too large, or the shape is 'strange' and the range of gray levels in the object is large enough, perform BCV thresholding in the object region. (4) Repeat the above step until no further object can be thresholded, according to these rules. Before formally presenting the algorithm, we will discuss data characteristics.

Characterization of data of one application example

Apart from the fact that the particles are locally brighter than the background, there are some characteristics of the grey value variation in the interior of the particles, which, generally-speaking, are fairly moderate. The images are taken with the sky as background. The sky, whether overcast or clear, is quite bright in comparison with the particles. A statistical analysis covering thousands of particles in long image sequences in our application (falling aggregate particles) shows that in most particles, the range of grey levels is less than 50. Some particles have a range around 80, often due to brighter spots around the edge of the particle, or, illumination differences on either side of interior edges separating two or more faces of a particle. Roughly speaking, there are three categories of imaged particles. For images of normal brightness the global threshold yields some kind of medium grey (around 128), and all objects have, by definition, grey values below this threshold. The first category are those particles for which grey values are in the range 40 to 85, and they are in a clear majority. The second category covers many fewer particles, and they tend to have a grey value range [40,85], except for 1-10% of the pixels in each particle. The third category may have up to half of the pixel grey values outside the range [40,85], and the other half within that range. When thresholding for the second class, this will result in the "loss" of some interior parts of the particle, or, of the particles appearing to break up into several pieces. To overcome this problem, we *define* Δf as the range when excluding 10% of the pixels, namely those with grey values from 0 to the value at which the *P-tile* is 5%, and the tail from the value at which the *P-tile* is 95% up to 255, *where P-tile* refers to the relative grey value histogram of the object, (The object can be a particle, or, a cluster of particles, or, a mix of background and particles.)

Formal algorithm description of an example

Let H_i *be the local gray value histogram of object i,* $i \in L^{(t)}$.

Define A_i *(area in pixels),* $S_i = P_i / 4\pi A_i$ *and* $\Delta f_i = 95\%tile\left(H_i\right) - 5\%tile\left(H_i\right)$.

Let BCV (region) denote BCV applied to a region.

Let $L^{(t)}$ *be the list of numbered objects in iteration t, t = 1,2,...*

```
BCV (whole image)
loop: {  calculate $A_i$, $S_i$, $\Delta f_i$ for $i \ni L^{(t)}$
         $\forall$  $i \ni L^{(t)}$  {
               if $A_i > 6400$ and $\Delta f_i > 50$ do {
                     BCV ($object_i$), obtaining $L^{(t+1)}$
```

```
            goto loop
                  } else
                  if $1225 < A_i < 6400$ and $\Delta f_i > 50$ do {
                  if $S_i > 2.8$ do {
                     BCV ( $object_i$ ), obtaining $L^{(t+1)}$
                     goto loop
                    }
                  }
         }
 }
```

Fig. 13. Pseudo-code of Algorithm.

When formulating the algorithm below, we let the range test $\Delta f > 50$, where 50 is a *range threshold*, be a basic criterion determining whether to go on with recursive thresholding in current objects or not. For large objects consisting of both background and particles, the range of grey levels need to be large, $\Delta f > 50$. For touching particles, forming an object (without background), the range (as defined by Δf) is often, but not always, small. For small particles (below 35 mm) the range criterion is normally not applicable (bright spots around the edge of the particle). This is an example of the third category mentioned in the previous paragraph.

Hence, after global BCV thresholding, each of the objects is labeled and area A and perimeter P are calculated. Provided the grey value range is sufficiently large (e.g. $\Delta f > 50$), BCV thresholding is applied to an object if it is really large, $A > 6400$ pixels, or, if it is sufficiently large, i.e. $1225 < A < 6400$ and of 'complex shape', where we use a simple shape factor $S = P^2 / 4\pi A$ [if more detail shape information is needed, the least squares oriented Feret box algorithm can be used], defining $S > 2.8$ as complex shape (from experience, about hundred images were tested visually and using an interactive program).

Examples

Figure 14 and Table 1 show one example, where Figure 14A illustrates an original image. After global BCV thresholding, the binarized image is given in Figure 14B. The largest white object is given a label number 0. Because the object no. 0 has an area greater than 6400 and the range of grey levels is 116, the algorithm does BCV thresholding in the corresponding area of the original image, and the binarized image is shown in Figure 14C. In Figure 14C object nos. 1,2, 6, and 9 have an area greater than 1225 pixels. Expect for the object no. 9, the other three objects had a shape factor $S > 2.8$, and where selected for further processing and tests. Out of these three objects, nos. and 2 had a range of gray levels over 50. At the end of the whole sequence, the algorithm performed BCV thresholding both for object nos. 1 and 2 in the original image, and the final result is displayed in Figure 14D. To see if the new algorithm works better than some standard local / adaptive thresholding techniques, in the next section we will compare two widely-used thresholding algorithms.