14

Estimation of Crystal Size Distribution: Image Thresholding Based on Multi-Objective Optimization

Karthik Raja Periasamy and S. Lakshminarayanan
Department of Chemical and Biomolecular Engineering, National University of Singapore, Singapore

14.1 Introduction

Crystallization is one of the basic unit operations employed in the pharmaceutical industries. The size, shape and purity of a crystal influences further downstream processing. Hence, it is critical to control the crystallization process. Control of the crystallization process is made more challenging because of its high sensitivity to disturbances [1]. Crystal size distribution (CSD) is one of the important characteristics to be monitored and controlled in order to obtain crystals of the desired quality [2]. Techniques such as laser based focus beam reflectance measurement (FBRM) and particle vision and measurement (PVM) are extensively used for online monitoring of the crystallization process.

In the FBRM technique, a laser beam is focused into the crystallizer using a rotating lens. The light is scattered when the beam passes through a particle. Based on the duration required for the light to scatter back, the chord length of the particle is measured. The major drawback in the FBRM technique is that the chord length distribution measured is not the actual particle size distribution because the chord length measured randomly may not represent the entire particle. This limitation can be overcome with the use of the

Multi-Objective Optimization in Chemical Engineering: Developments and Applications, First Edition.
Edited by Gade Pandu Rangaiah and Adrián Bonilla-Petriciolet.

PVM technique as direct measurement from the process image is possible [3]. In the PVM technique, the process images are captured either as videos or pictures with the help of a camera; these pictures are analyzed to obtain a direct estimate of CSD [4].

Offline based image analysis was initially used for the estimation of CSD [5, 6, 7, 8, 9, 10]. However, the need for automated online methods arose. In 2006, an algorithm for segmenting high-aspect ratio particles using linear feature detection was developed [11] . Later, this algorithm was extended to detect other shapes but was restricted to typical geometric shapes only [12]. Zhou *et al.* developed an image analysis algorithm using multivariate statistics based image analysis coupled with standard image processing techniques [4, 13]. The drawback with this method was that the algorithm was restricted to detecting square and diamond shaped particles and could result in inaccuracies because of the random orientation of particles in the solution. Korath *et al.* used the change in intensity to identify touching particles [14, 15]. The measurements using this method suffered due to morphological image processing. Mironescu *et al.* made use of fractal analysis to compute CSD [16]. Preseles *et al.* designed an algorithm by improving an adaptation of the watershed segmentation method [3]. However, the computational effort of the algorithm was high. In this work, a new multi-objective based image segmentation segmentation method is used for the estimation of CSD.

The proverb "an image is worth a thousand words" says it all. Perhaps a picture is worth several thousand data samples for it can best reflect the actual state of some processes. Image analysis plays a crucial role in extracting meaningful and actionable information from process images. The human eye and the brain together are the best example of an image analysis system. Computers can be used to replace human effort so as to make the image analysis process much more fast, efficient and automatic. Image analysis is applied in diverse fields such as biology, medicine, remote sensing, robotics and manufacturing. Image segmentation is crucial in image analysis because the quality of segmentation plays a vital part in feature extraction.

Image segmentation is defined as the process of extracting meaningful information from an image by dividing the image into multiple segments. This step is carried out on the gray level values of the image. Image thresholding is a basic and easy-to-implement technique. This technique partitions an image into regions based on a predefined criterion. The challenging aspect in thresholding is to choose an optimal threshold. Many techniques can serve the purpose – one such technique is based on the image histogram. An image histogram is defined as the frequency distribution plot of gray level intensities. Single-objective optimization (SOO) is a method used to select the optimal threshold. Several different methods use SOO to select an optimal threshold. Some commonly practiced methods are the Otsu method, the minimum error method, the mode method, and the entropy method [17, 18]. Each of these methods has its own limitations. The limitations of different methods can perhaps be overcome by combining more than one objective. Hence, multi-objective optimization (MOO) is utilized in this work to identify a suitable threshold. This approach can assist in improving the accuracy of CSD estimation thereby leading to efficient control of the crystallization process and consequently attaining the desired product quality. The novelty in this work is that a MOO based thresholding approach has not been applied previously to solve problems such as segmentation of crystals from process images. This work has been further extended to estimate the CSD in two dimensions (length and width) and the effect of particle orientation in the solution [19].

The outline of this chapter is as follows. In section 14.2, the image analysis steps used in this work are explained briefly. Then, artificial image simulation is discussed with an example in section 14.3. Section 14.4 explains the techniques used in preprocessing of images. Image thresholding, using different methods to select thresholds based on SOO, is discussed in section 14.5. The need for MOO is explained and the approach used in this work to solve the MOO formulation is outlined. The threshold identification problem is presented as a two-objective optimization problem and applied to the simulated images. Section 14.6 explains the feature extraction technique and the crystal size distribution (CSD) estimation results. Sections 14.7 and 14.8 discuss future work and conclude the chapter.

14.2 Methodology

The steps used for obtaining the CSD from the process images are shown in Figure 14.1.

The PVM images obtained from the camera are used for estimation of CSD. Hence, image preprocessing is carried out initially to improve the image quality by noise reduction and contrast enhancement. Then, image segmentation is applied to extract the region of interest (crystals) from the background of the image. This is carried out by partitioning the image into segments. Finally, with the help of feature extraction, the given image is transformed into a set of features. In this work, the characteristic dimension (length) of the crystal is extracted as its feature. The characteristic length of the crystals in the crystallizer image is considered to represent the CSD of the system. In the work presented here, MATLAB 7.11.0 (R2010b) was used for image analysis.

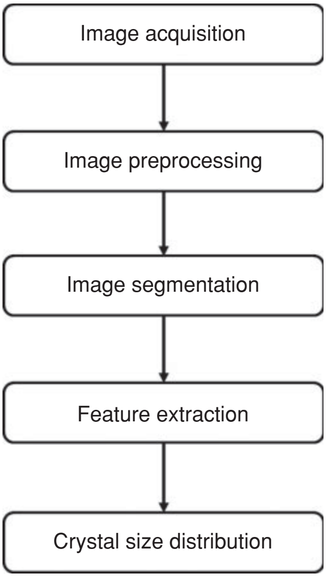

Figure 14.1 *Steps in image analysis for a crystallization process.*

14.3 Image Simulation

Artificial images were simulated randomly by considering an appropriate camera model to depict the adapted process model.

14.3.1 Camera Model

The method of mapping points from a real-world three-dimensional plane onto an imaginary two-dimensional plane is known as projection. When the human eye looks at a scene, objects farther away appear smaller than the objects close by—this property is known as perspective. Orthographic projection is a projection method where the object is mapped onto an imaginary plane without considering the distance between the object and the observer. Orthographic projection is used to create images to scale drawings. This projection allows accurate measurements to be made. Weak perspective projection can be defined as a perspective projection approximated by a scaled orthographic projection. This projection is considered when the object's dimensions are negligible compared to the distance between the object and the camera as shown in Figure 14.2 [20].

14.3.2 Process Model

The first step is to define a process model bearing in mind a certain number of assumptions. In this work, the particles are assumed to be of high aspect ratio (acicular or needle shaped). Generally, needle-shaped crystals are observed in crystallization processes involving monosodium glutamate, calcium pyrophosphate dehydrate, monosodium urate, and so forth. Solid phase particles dispersed in a crystallization solution bath are assumed to be the process model. Let us assume a volume, V, to be the volume of the image focused by the camera. Assume n solid particles are present in volume V. Let the characteristic length of every particle in the given volume be denoted by l. Hence, the CSD in the imaging volume is given by a function f(l).

In this process model, the particle population is simulated by randomly generating several parameters of the particle such as the position, orientation, and dimension. The orientation of the particle in the image plane is distributed normally. The dimensions and position of the particle in the image plane are distributed uniformly for illustrative purposes. The parameters are generated randomly independent of each other.

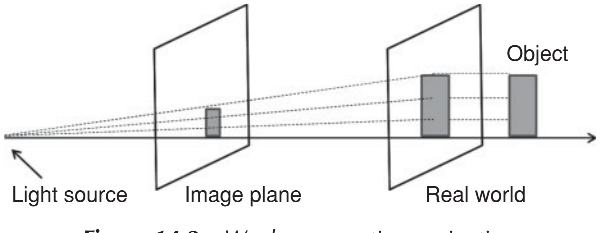

Figure 14.2 Weak perspective projection.

The transformation from a 3D point *X, Y* and *Z* in real-world coordinates to an image point x_I and y_I is given by Equation (14.1), where *f* denotes the camera's focal length:

$$x_I = \frac{fX}{Z}; y_I = \frac{fY}{Z} \tag{14.1}$$

14.3.3 Assumptions

The real-world coordinates are mapped into coordinates on the image plane. It is assumed that the depth of the imaging volume is negligible compared to the fixed distance between the camera and the imaging volume because the distance of the camera from the particle will be greater than the depth of the particle. Hence, a weak perspective projection model is used to project the imaging volume onto the image plane. The crystal particles are scaled by a constant ratio.

Generally, the crystallization process is assumed to take place in a well-stirred vessel. The crystals are therefore assumed to align themselves parallel to the surface of the crystal solution because of the shear force acting between the surfaces of the crystal and the solution. The particles are therefore believed to be orthogonal with respect to the camera's optical axis and thereby mimic the actual condition used for imaging measurements. A comparable methodology was used in [21] to simulate artificial crystallization images. However, in contrast to the work presented in [21], in this study, the gray level values of the foreground (crystals) and background are also stochastically simulated. Also, more importantly, in our work, noise is incorporated in to the images generated so as to mimic images obtained from industrial crystallization processes.

The intensities of the slurry and the particles were generated randomly independent of each other and distributed uniformly. Particle overlaps were also added to the system stochastically. Multiplicative noise was incorporated in the simulated image. Some broken crystals and crystal chips were also added into the image. Imaging volume is taken to be large enough so that all the particles generated stochastically are contained within the imaging plane itself. Hence, the position and dimension of the particles are confined so that the simulated particle remains within the imaging plane. An image simulated using the above algorithm and an actual image from a crystallization process [4] are shown in Figure 14.3.

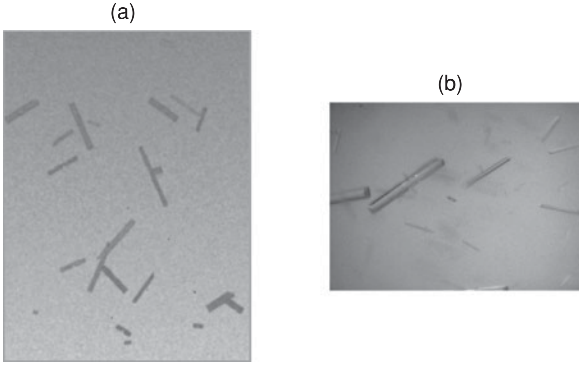

Figure 14.3 (a) Artificial image; (b) real-world image.

14.4 Image Preprocessing

Generally, a digital image is represented as a three-dimensional matrix indicating each of the basic colors red, green, and blue. These images are first converted to single dimensional matrix for further processing. The single-dimensional matrix represents the gray level intensity at each pixel. The gray level intensity matrix *GL* is computed as the weighted sum of the three matrices (red *R*, green *G*, and blue *B*) as shown in Equation (14.2)[22]:

$$GL = 0.2989 * R + 0.5870 * G + 0.1140 * B \tag{14.2}$$

The objectives of preprocessing include noise removal and image enhancement. The grayscale images contain a considerable amount of noise. Noise in the images can originate due to the sensitivity of camera to light and/or during data transfer and storage (file formats). Errors in data transmission can cause black and white specks randomly throughout the image, commonly known as impulse noise. Camera sensors are prone to cause noise because of their inability to differentiate between the photoelectric effect electrons generated by the heat produced in the system and the electrons generated by the actual signal [23]. The effect of this type of noise is generally dependent on the input signal. Hence, the noise is assumed to be multiplicative in nature. This type of noise is known as speckle noise. Filtering techniques are commonly used to remove noise. Different filtering techniques are used to address different types of noises [24, 25]. This type of noise cannot be removed with the help of a linear filter. Therefore, a type of nonlinear filter called the median filter is applied for removing noise from the images considered. This type of filter is useful in removing impulse and speckle noise. It has the additional advantage of preserving the edges of objects and is very apt for the current application.

The other purpose of preprocessing is to enhance the quality of the image [24, 25]. Image enhancement is accomplished by adjusting the contrast in the image. Contrast enhancement is applied to images to widen the range of the gray level intensity. This is done by mapping the intensity values of the original image into new values by scaling the lower and upper bound to 0 and 255, respectively.

14.5 Image Segmentation

14.5.1 Image Thresholding Based on Single Objective Optimization

In image segmentation, the preprocessed image is converted into a binary image with a user-specified threshold. A binary image is considered to represent two classes in an image, namely the object and the background. Pixels with intensity above a certain threshold value are categorized into one class and those below the threshold into another class. The threshold value required for image segmentation can be selected based on various methods. The mode method is the most basic method [18]. It uses the concept of a valley to identify the threshold. In this method, the point at which the intensity value is lowest between the two class of pixels in the image histogram is selected to be the threshold. In the entropy method, the optimal threshold is selected so that the sum of the entropy of the two classes (foreground and background) reaches its maximum [26]. The Otsu method uses the concept of maximizing the variance between the two sets of pixels, that is the object and the background [27]. The minimum error method [28] fits Gaussian curves on the image using the threshold value. In this method, two Gaussian curves are fitted to the two peaks (foreground and background) on the image histogram. The intensity value at which one Gaussian curve ends and the other begins is taken as the threshold value. The threshold value where the error in fitting is minimum is the optimal threshold. The individual objectives in the MOO are described next.

14.5.1.1 Otsu Method

Otsu developed a method for choosing a suitable threshold value by maximizing the interclass variance [27]. This method involves calculating the variance of the two classes of pixels at all possible threshold values. The aim of this method is to separate the pixel classes as far as possible. The optimum threshold is computed by maximizing the interclass variance.

In any grayscale image, n_i denotes the number of pixels at a given gray level value i. N and L denotes the total number of pixels and gray levels respectively. An image histogram is constructed and normalized to a probability distribution as shown in Equation (14.3):

$$P_i = n_i/N \tag{14.3}$$

Equations (14.4) and (14.5) are used to calculate the histogram's zeroth and first-order cumulative moments up to the k^{th} gray level whereas Equation (14.6) computes the total mean level of the original picture:

$$\omega(k) = \sum_{i=1}^{k} p_i \tag{14.4}$$

$$\mu(k) = \sum_{i=1}^{k} ip_i \tag{14.5}$$

$$\mu_T = \mu(L) = \sum_{i=1}^{L} ip_i \tag{14.6}$$

Equation (14.7) gives the function to be maximized in order to calculate the optimal threshold, k:

$$\sigma^2 B^{(k)} = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} \tag{14.7}$$

The k value has a constraint such that $1 < k <$ L.

14.5.1.2 Minimum Error Method

In this method, a classification approach is used to select a suitable threshold. If the gray-level distribution of the object and background can be estimated, the minimum error based threshold can be obtained. The assumption used in this method is that the estimated object and background populations are normally distributed. Based on this assumption, Gaussian curves can be used to fit the gray level distributions of the foreground and the background by estimating their parameters (mean, standard deviation and probability) from which the optimum threshold can be obtained.

This method was applied by Kittler and Illingworth [28] to calculate the threshold. In their work, $h(g)$ is used to represent the image histogram. Then the image is segmented at a random threshold T resulting in two distributions. The two distributions (object and background) are modeled to fit a normal distribution by estimating parameters $\mu_i(T)$ and $\sigma_i(T)$ and *a priori* probability $P_i(T)$. The parameter values are given in Equations (14.8), (14.9) and (14.10):

$$P_i(T) = \sum_{g=a}^{b} h(g) \tag{14.8}$$

$$\mu_i(T) = \left[\sum_{g=a}^{b} h(g)g\right] \Big/ P_i(T) \tag{14.9}$$

$$\sigma_i^2(T) = \left[\sum_{g=a}^{b} \{g - \mu_i(T)\}^2 h(g)g\right] \Big/ P_i(T) \tag{14.10}$$

where a and b are calculated by Equations (14.11) and (14.12), respectively

$$a = \begin{cases} 0 & i = 1 \\ T + 1 & i = 2 \end{cases} \tag{14.11}$$

$$b = \begin{cases} T & i = 1 \\ n & i = 2 \end{cases} \tag{14.12}$$

The optimum threshold is obtained by minimizing the function described in Equation (14.13). This equation computes the probability of misfit by comparing the actual gray level distribution with the assumed object and background populations:

$$\begin{aligned} J(T) = 1 + 2[P_1(T)\log\sigma_1(T) + P_2(T)\log\sigma_2(T)] \\ -2[P_1(T)\log P_1(T) + P_2(T)\log P_2(T)] \end{aligned} \tag{14.13}$$

The threshold is calculated by finding a value for T such that the function described by Equation (14.13) is minimized.

14.5.2 Multi-Objective Optimization

The threshold is often calculated by solving a SOO problem [18]. The segmentation performed based on the methods mentioned above works well for many cases. However, each method has its own limitations. The Otsu method fails in images where the two clusters of pixels are of different sizes. This results in the objective function attaining multiple maxima and thereby increases the chances of selecting a local optimum [29]. The drawback of minimum error thresholding occurs when the peaks in the histogram are very close to each other. Such a situation results in errors during the estimation of parameters from the truncated distribution [30]. The weakness of the individual methods can be improved by applying a hybrid approach involving both the objectives to improve the segmentation process. Hence, a multi-objective optimization problem can be solved to obtain an optimum threshold [31, 32, 33].

Multi-objective optimization (MOO) is a method used for solving problems that have more than one conflicting objective under certain constraints. It has applications in many domains including product design, medical settings, and the automobile, oil, and gas industries [34]. It is performed where decisions are to be taken by a tradeoff between the conflicting objectives. Upon solving the MOO problem, a set of solutions known as the Pareto front is obtained. The user can choose an optimal point from the Pareto set using *a priori* knowledge about the problem. A Pareto front is defined as a set of solutions that are non-dominated to each other—no solution in the set can replace another solution in the set to improve an objective without worsening another objective. For example, let us consider a MOO problem with two conflicting objectives, f_1 and f_2, that are to be minimized. Upon solving the problem, we can obtain a solution vector $x = [x_1, x_2, x_3, \ldots x_\mathrm{n}]$. A solution x_i is said to be dominating another solution x_j, if the conditions $f_1(x_\mathrm{i})<f_1(x_j)$ and $f_2(x_i)<f_2(x_j)$ hold true. A solution x_i is said to be a Pareto solution, if there doesn't exist any solution x_j such that it dominates x_i within the solution space [35, 36].

There are several techniques in the literature to solve MOO problems. Techniques used for obtaining Pareto-optimal solutions include the weighting method, the ε-constraint method, and non-dominated sorting genetic programming (NSGA-II). The first two approaches transform the MOO problem into a SOO problem and solve the problem using SOO problem solving methods [37]. NSGA-II is an evolutionary algorithm, inspired by the natural evolution process [38]. In this method, a population of solutions is generated stochastically and then this population is evolved through genetic operations into to a more appropriate solution over several generations.

14.5.2.1 Converting a MOO Problem into a SOO Problem

In this work, the weighted-sum approach is applied to solve the MOO problem. The weighted-sum method is the simplest method to solve a MOO. This method can be used to solve a MOO when the solution is convex [37]. In this approach, the set of objectives is scalarized into a single objective by multiplying each objective with the help of a certain weight and aggregating the objectives together. The formulation of the weighted-sum approach is given by:

$$\begin{aligned} &minimize \quad \sum_{i=1}^{k} w_i f_i(x) \\ &subject \quad to\, x \in S \end{aligned} \tag{14.14}$$

The conditions for applying this method are that objectives are scaled to bring them to the same magnitude and weights are assigned non-negative values ranging between 0 to 1, that is, $wi \epsilon$ [0,1].

Summation of all importance factors is equal to 1, that is

$$\sum_{i=1}^{k} w_i = 1.$$

For mixed optimization problems *(min-max),* we need to convert all the objectives into a single type.

A Pareto front can be obtained by solving the SOO for different values of the weights. The weights are also known as importance factors as they measure the importance of each objective in the process. The problem is solved for many discrete values of weights between [0, 1]. The solution obtained is considered Pareto optimal if the weights are positive. Based on the user's requirements, he or she can choose an optimum from the Pareto set. In our work, we have used simulated annealing to solve the SOO problem obtained by converting the MOO problem.

14.5.2.2 Simulated Annealing

Simulated Annealing (SA) is a widely used search algorithm to detect global optima in the feasible region [39]. The statistical method initially developed to model the metal annealing process by Metropolis *et al.* [40] has been the inspiration for the simulated annealing algorithm, which is now used to tackle discrete optimization problems [39]. Annealing is a well studied physical process used to reduce deformities in a metal. It is typically done by heating the metal to melting temperatures to mobilize atoms, holding the temperature for a while, and then cooling it back to obtain the necessary rigidity. Repeated cycles of heating and cooling help atoms trapped in local energy minima to sample wider spaces and achieve global low energy states. The algorithm developed by Metropolis *et al.* calculates new energy values at each step and compares these to the current energy values. If the new energy value is found to be lower, this new state is accepted. If the new energy value is higher than the current value, the state is not rejected outright but is accepted based on a probability that has an inverse relationship to the difference in energies. At any temperature *t,* the probability function is given by

$$p(\delta E) = \exp\left(-\frac{\delta E}{kt}\right) \tag{14.15}$$

where k is the Boltzmann constant. This process of finding new states is iterated until the system achieves the required low energy states or until certain conditions are met. Kirkpatrick *et al.* [41] and Cerny [42] separately used this method to solve optimization problems by studying the metal annealing process.

Local optimization algorithms like the hill-descent method depend on initial guess values to arrive at the final local optima. Simulated annealing outperforms these methods because the uphill moves in this algorithm are systematically generated based on energy differences, whereas randomness is involved in the selection of uphill movements in the other methods (15).

The algorithm used in simulated annealing is described below:

1. Choose a starting state s_0. Compute the energy of the state $f(s_0)$.
2. Choose a temperature $t_0 > 0$.
3. Choose a cooling rate α.
4. Choose a new state stochastically from the possible moves from s_0. Compute the energy of the new position, $f(s)$.
5. If $f(s) < f(s_0)$ let $s_0 = s$. Otherwise simulate a random number x uniformly distributed in the domain [0, 1]. Verify the condition if $x < \exp((f(s_0) - f(s))/t)$, let $s_0 = s$. Otherwise do not perform any moves. Iterate steps 4 and 5 untill a maximum limit on the number of iterations is reached.
6. Reduce the temperature at the given cooling rate α and repeat steps 4 to 6 until a steady state is attained—s_0 is approximated to be the optimum.

14.5.3 Problem Formulation

The aim of the optimization process here is to find the optimum threshold value that can be used to best segment the gray level image. Two objectives are considered here. The first objective is to maximize the interclass variance and the second objective is to minimize the error. This is a mixed optimization problem. Therefore, the objective to be maximized is changed appropriately to be minimized – the problem required to be minimized is therefore formulated as in Equation (14.16):

$$F(t) = w_1 * \frac{1}{\sigma_B^2(t)} + (1 - w_1) * J(t) \tag{14.16}$$

where σ_B and J are computed from Equations (14.7) and (14.13) respectively.

Both the objectives are normalized to a similar magnitude in order to avoid biasing the main function. The search space varies from 0 to 255 indicating the pixel values in the image. Although this is a combinatorial optimization type of problem, exhaustive enumeration cannot always be used to search the complete solution domain due to the large computational effort required. Simulated annealing is therefore applied to determine the global optimum. Due to the annealing process, this method can find the global (or near global) optimum for problems that have numerous local optima. The optimization is carried out by varying the weight w_1 from 0 to 1. Since the Pareto-optimal front for this problem is convex, the weighted-sum method gives the Pareto frontier.

Once a Pareto front is obtained, an optimal solution can be selected based on *a priori* knowledge. As we do not have any *a priori* knowledge to select an optimal solution, post-Pareto optimality analysis is carried out using the L_2-norm method to find an optimal compromise solution [43]. This method uses the concept of a utopia point to find the optimal solution as described in Figure 14.4. A utopia point is defined as the theoretically best achievable point and is assumed to be the origin in most cases. In our approach, the L_2-norm method finds the solution that is geometrically nearest to the utopia point (assumed to be the origin).

The segmentation quality is evaluated by calculating the error in pixel classification. The error in pixel classification is found by calculating the sum of number of pixels wrongly classified (background pixels classified as object pixels and object pixels classified as background pixels) [44].

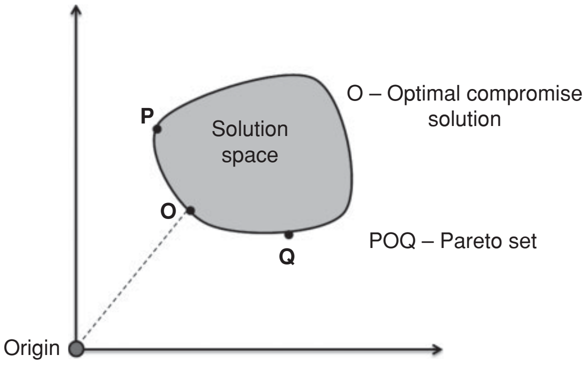

Figure 14.4 Optimal compromise solution based on the L_2-norm method.

After image segmentation, morphological image processing is applied to binary images. Morphological operations are performed to change the structure of the objects based on user requirements such as removing undesired objects like broken particles or noise, and identifying the outlines of objects [25]. They are used for representation of image shapes. There are two fundamental morphological image operations known as dilation and erosion. The erosion operation removes the boundary particles and, hence, the skeleton of the object is obtained. In the dilation operation, the object grows or thickens. The boundary of the objects enlarges to allow edges to be continuous.

14.5.4 Results and Discussion

The MOO based segmentation explained above, along with morphological operations, were tested on several images. Two examples are shown and the results are discussed. The preprocessed images were segmented by three different methods (i) the Otsu method, (ii) the minimum error method, and (iii) multi-objective optimization based segmentation method. The results are compared. In the work presented in this chapter, Otsu and minimum error methods were used without any modifications.

14.5.4.1 Example 1

In the first example, we consider an image with few crystals, considering that the process is in the initial stages. The MOO problem is solved for different values of weight w_1.

The Pareto front is obtained by plotting the minimum error versus the inverse of the interclass variance on the logarithmic scale. From Figure 14.5, it can be observed that

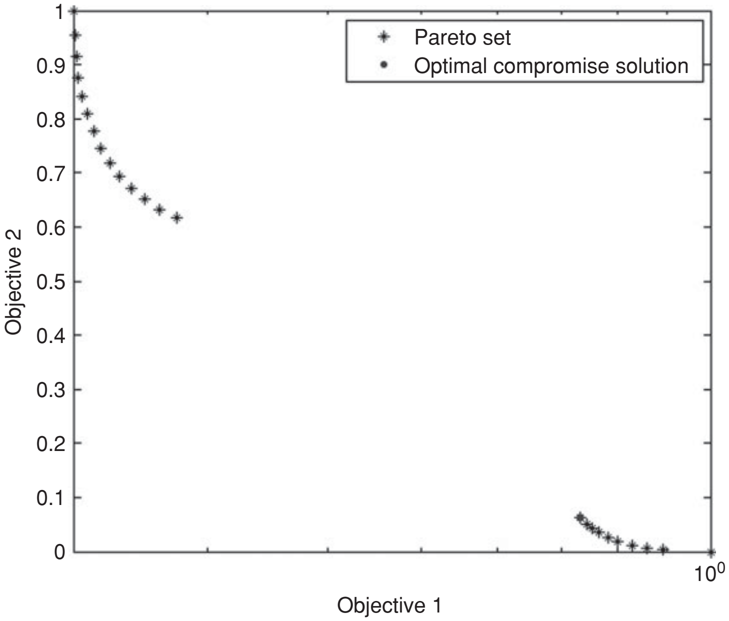

Figure 14.5 *Pareto plot.*

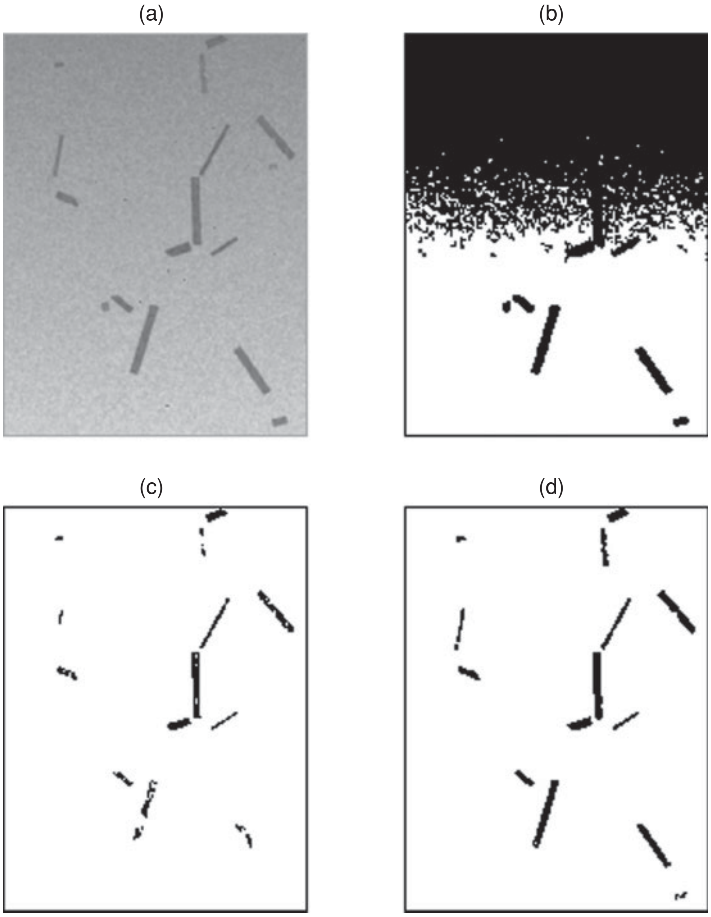

Figure 14.6 *Example 1 (a) - Original image, (b), (c), (d)—Image after thresholding using the Otsu method, minimum error method, and MOO based segmentation respectively.*

there is a gap in the Pareto front, which shows that there is no non-dominated solution in that region. The optimum threshold value is obtained by using the L_2-norm method. This threshold is used for the segmentation of the process image shown in Figure 14.6(a). The images obtained after thresholding using the Otsu and minimum error methods are shown in Figures 14.6(b) and 14.6(c), respectively.

The image obtained by thresholding using the optimal threshold calculated by the MOO method is given in Figure 14.6(d). From Figure 14.6(b), it is clear that the Otsu method segments the particles of darker intensity clearly, but, while dealing with lighter intensity particles, it fails to differentiate between the background and the object. Similarly, from Figure 14.6(c), it can be said that outlines of most particles are identified but the complete shapes of the particles are not characterized properly. In Figure 14.6(d), it can be seen that MOO based thresholding overcomes the limitations of the other methods and identifies most particles completely. This comparison can be further verified by calculating the

Table 14.1 *Misclassification rate.*

Image name	Otsu method	Minimum error method	MOO based segmentation
Example 1	48.38	1.73	0.40
Example 2	37.15	5.58	1.34

misclassification rate of the segmentation method. The misclassification rates of the three thresholding methods are shown in Table 14.1. From row 1 of Table 14.1, it is clear that there is less misclassification with the MOO approach than with the Otsu or minimum error method.

14.5.4.2 Example 2

In the second example, the number of particles in the image is increased. This is to simulate the later stage of a crystallization run during which we expect to have more crystals in the crystallizer.

The Pareto front (shown in Figure 14.7) is obtained similar to Example 1. Optimal threshold obtained using the L_2-norm method is used for the segmentation of the process image shown in Figure 14.8(a). The images obtained after thresholding using the three considered methods (Otsu, minimum error and the MOO based approach) are shown in Figures 14.8(b), (c) and (d) respectively. As in example 1, it is clear that Otsu method based thresholding fails to differentiate between the particle and the background while minimum error based thresholding does not capture the particle contour as well as one might have hoped. From Figure 14.8(d), it can be seen that MOO based thresholding performs better

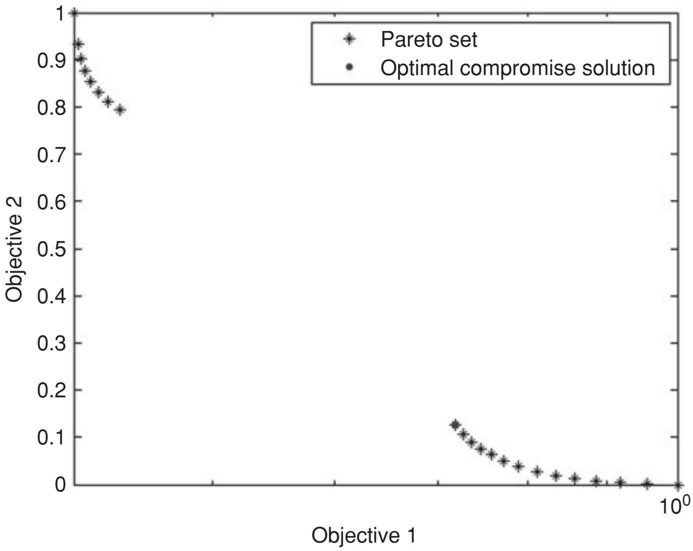

Figure 14.7 *Pareto plot.*

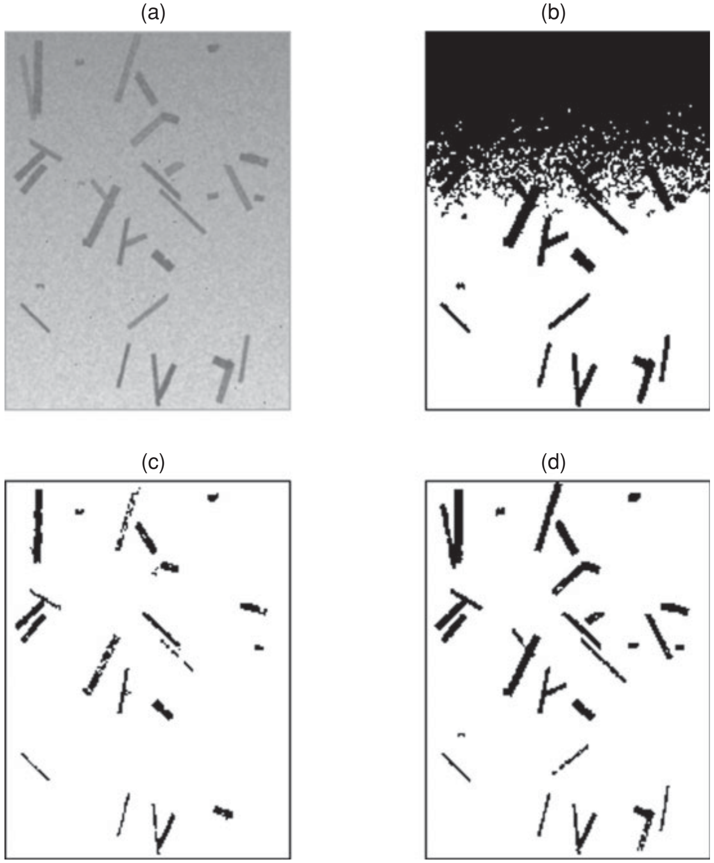

Figure 14.8 Example 2 (a) Original image, (b), (c), (d) Image after thresholding using Otsu method, minimum error method, and MOO based segmentation, respectively.

than the other two methods. The performance is validated using the misclassification rates as indicated in the last row of Table 14.1.

From the results shown in Table 14.1, it can be noticed that misclassification rate of the Otsu method is much higher compared to the minimum error method and the MOO based thresholding method. The misclassification rates confirm that minimum error method performs better than the Otsu method, while MOO based thresholding outperforms both the methods. From Table 14.1 we infer that, as expected, the misclassification rate increases as the number of particles in the image increases.

14.6 Feature Extraction

The characteristic length of each segmented object has to be calculated. Hence, the feature extraction step is used to extract features of the region of interest. The technique employed

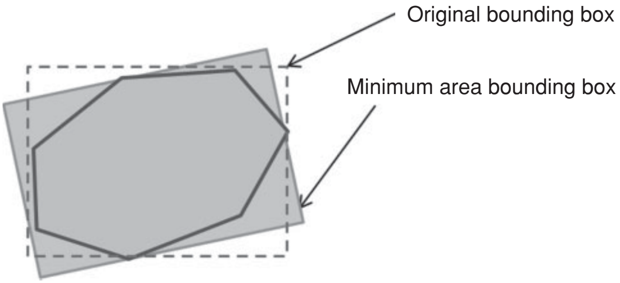

Figure 14.9 *Minimum area enclosing rectangle.*

to perform feature extraction is blob analysis [45]. In the binary images, the foreground is denoted by 1 and the background is denoted by 0. The pixels that have same values and are connected (touching particles) are referred to as blobs. Separate objects are referred to as different blobs in the image. Using blob analysis, one can find numerous statistics such as the number of blobs, position of the blobs, and the area of each blob.

In blob analysis, the minimum area rectangle algorithm is used such that the bounding box encloses the polygon (shown in Figure 14.9). The features of this bounding box are taken as the properties of the blob. The minimum area rectangle is found using the rotating calipers method [46] as described in algorithmic form below.

1. Find four extreme points for the polygon.
2. Draw support lines parallel to both x and y axes passing through the extreme points ensuring that the lines form a rectangle. These lines make a certain angle with different sides of the polygon.
3. Rotate the support lines in a clockwise direction so that one of the lines coincides with an edge of the polygon.
4. Compute the area of the newly formed enclosure and compare it with the stored area. If the new area is less than the stored area, take the new area as the minimum.
5. Repeat steps 3–4 until the total rotation of the lines is greater than 90 degrees.
6. From this method, the minimum area rectangle is obtained.

14.6.1 Results and Discussion

With the help of blob analysis, the CSD was estimated from the process images. This method was tested on numerous images and is illustrated using the image used as Example 2 above. The example image is given in Figure 14.10(a). The image after segmenting using the MOO approach undergoes morphological processing and is given in Figure 14.10(b). The image containing the extracted particles is given in Figure 14.10(c).

The accuracy of the proposed method was found by comparing the estimated CSD with the true (known) particle characteristics (true CSD) used to generate the artificial images. The estimation accuracy in an image is calculated by computing the sum of absolute error between the actual length and the characteristic length of each particle estimated by the algorithm. The estimation accuracy was tested on different sets of images by changing the number of particles in the process images. The algorithm was tested for three different sets

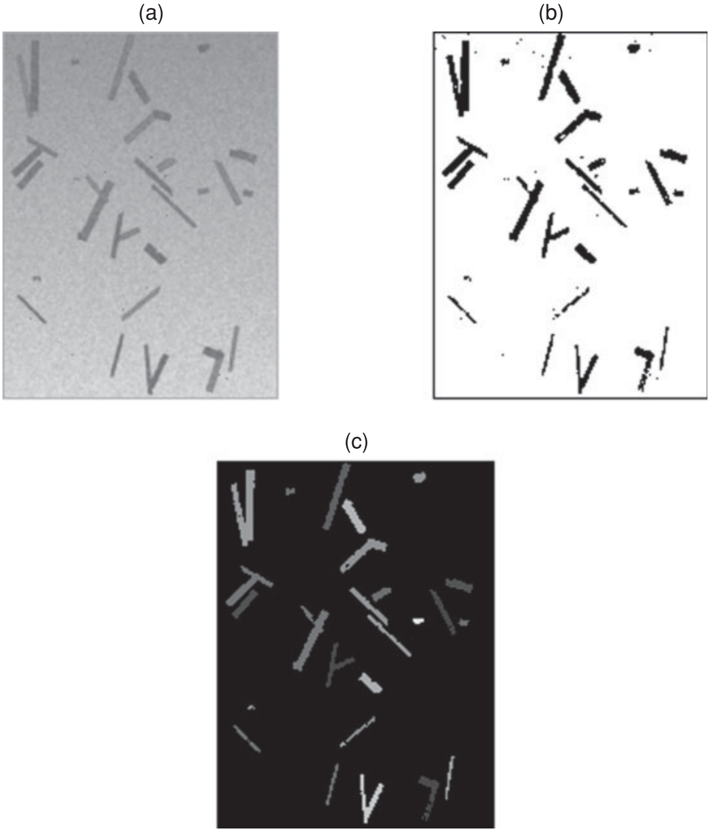

Figure 14.10 (a) Original image, (b) segmented image, and (c) final image.

of 100 images representing different size ranges. The results are compiled in Table 14.2. The last three columns presented in Table 14.2 give the mean, maximum and minimum estimation accuracy with respect to the number of particles that exist in each image (given in first column). As the number of particles increased, it was noted that there is an increase in the percentage of overlapping particles in the image. Hence, the percentage overlap was calculated for each set of images as the ratio of number of particles that are overlapped

Table 14.2 Estimation accuracy.

		Estimation accuracy		
No. of particles	Average percentage overlap	Average	Maximum	Minimum
5–15	22.86	95.51	99.16	86.28
15–20	37.24	93.33	98.68	85.31
20–25	43.48	92.72	98.18	83.18

in the image to the actual number of particles and is presented in the second column in Table 14.2. The mean accuracy of the proposed MOO based algorithm was found to be higher than 95% when 5–15 particles exist in an image. The mean accuracy dropped down to around 92.7% when the number of particles in the image increased to 20–25 particles. Hence, it can be seen that the particle-size characteristics are captured very well by this method in each image. In this work, the overlapped particles have not been taken into consideration for calculating the estimation accuracy.

In PVM, a set of images is captured by the camera at successive time points until enough information exists to estimate the CSD, which is crucial for effective control of the crystallization process. To study the consequence of the number of processed images on the CSD estimation accuracy, a database of 500 particles was constructed and considered to be suspended in the crystallization solution. From this library, a set of 50 images in which 0 to 25 particles exist were generated, stochastically imitating the images captured by a camera during a crystallization process. By applying the algorithm developed in this work on these images, a crystal size distribution was constructed. This crystal size distribution was compared with the distribution of the actual library used to generate the images. This is shown in Figure 14.11, which shows that the estimated frequency distribution (estimated CSD) is very similar to the true CSD represented by the library data. The mean and the variance of the library of particles was 100.52 (in pixels) and 377.85 (in pixels^2) respectively while that of the estimated CSD are 101.63 (in pixels) and 427.70 (in pixels^2).

The proposed method was applied to the situation where the number of images captured to 100. The estimated CSD is plotted along with the true CSD in Figure 14.12 for comparison.

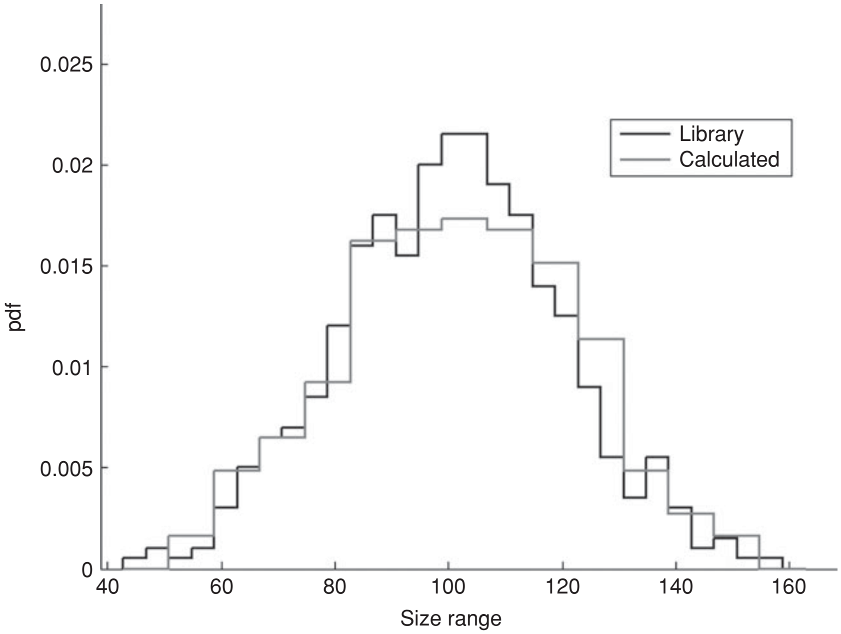

Figure 14.11 *Estimated crystal size distribution compared with model data (50 images).*

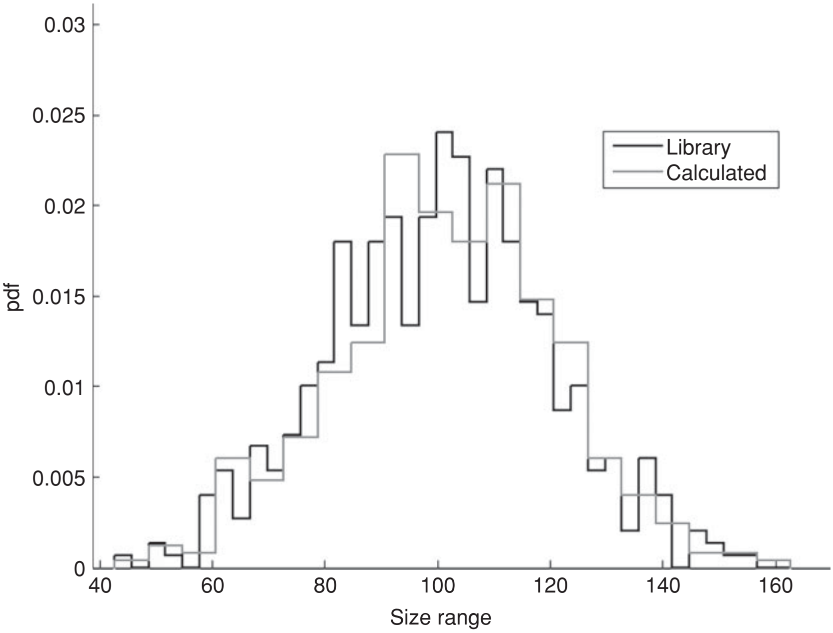

Figure 14.12 *Estimated crystal size distribution compared with model data (100 images).*

The mean and the variance of the new distribution are 101.46 and 377.66 respectively. From Figures 14.11 and 14.12, it was noted that as we increase the number of images captured, the coherence between the estimated CSD and the actual CSD improved further. It was also observed that the variance of the estimated distribution is closer to the actual variance of the distribution. This methodology can therefore be applied to images from the crystallization process for estimating CSD in order to control the crystallization process better.

14.7 Future Work

The proposed MOO based method has an average estimation accuracy greater than 92% when 20–25 particles exist in the image. However, this method does not estimate the size of overlapping particles. If the size of the overlapped particles can be characterized, the number of images required to obtain the CSD can therefore be reduced to bring down operation costs.

Therefore, the overlaps have to be handled directly during the processing of individual images. Overlapping particles need to be identified first. This can be done by using the minimum area rectangle method. Firstly, the minimum area enclosing the rectangle is calculated for the overlapping particle and compared with the actual area of the particle occupying the image obtained from blob analysis. Based on a small error criterion between the two areas, the blobs can be classified into overlaps and individual particles. After this step, there are two possible plans for separating the particles.

One method is to classify the overlapped particles into individual particles. A possible method to classify each pixel in this overlapped region is by assigning the pixel a label based on a set of rules. Second method is using model based segmentation, where models of particles can be fitted to separate touching and overlapping particles.

14.8 Conclusions

The estimation of CSD via image analysis is important for effective control of crystallization processes. The accuracy of the image analysis results largely depends on the image analysis methodologies chosen for the various image processing steps. Image segmentation is vital in the overall image analysis procedure. Image thresholding by traditional methods fails if the image is noisy, even after image preprocessing. However, the limitations of the traditional method can be overcome using the MOO based thresholding approach, as this chapter has shown. The CSD can be estimated using feature extraction techniques like blob analysis, and the minimum area enclosing rectangle. The results show that the proposed algorithm has high estimation accuracy owing to its MOO based thresholding approach. This technique, therefore, offers an opportunity for automated control of crystallization processes leading to improved product quality. Image analysis techniques can also be extended in the case of other particulate process involved in the pharmaceutical industry.

Abbreviation

CSD	crystal size distribution.
FBRM	focus beam reflectance measurement.
MOO	multi-objective optimization.
NSGA	non-dominated sorting genetic algorithm.
PVM	particle vision and measurement.
SA	simulated annealing.
SOO	single objective optimization.

Nomenclature

A	gray level range of the first Gaussian curve.
B	gray level range of the second Gaussian curve.
f	focal length of the camera.
$f_i(x)$	i^{th} objective function.
$f(l)$	estimated crystal size distribution.
$f(s)$	energy of the current state.
$f(s_0)$	energy of initial state.
$h(g)$	normalized distribution of the image histogram where g denotes the gray level.
i	gray level.
j	number of objectives.
k	Boltzmann's constant.

l	characteristic length of each particle in the given volume.
n	number of particles captured in the image.
n_i	number of pixels at each gray level.
p_i	normalized probability distribution of the image histogram.
s	current state.
s_0	initial state.
t	temperature of the current state in simulated annealing algorithm.
w_i	weight parameter.
x	solution vector.
x_i, y_i	coordinates on the image plane.
$F(t)$	total objective function.
GL	gray level values of each pixel in an image.
$J(T)$	second objective function.
L	number of gray levels in the image.
N	total number of pixels.
$P_i(T)$	probability of the i^{th} Gaussian curve.
V	volume of the process model to be captured by the camera.
S	search space.
T	optimal threshold.
X, Y, Z	physical coordinates.
α	cooling rate.
δE	change in energy.
$\mu_i(T)$	mean of the i^{th} Gaussian curve.
$\mu(k)$	first-order cumulative moments of the histogram up to k^{th} gray level.
μ_T	total mean gray level of the image.
$\sigma^2{}_B(k)$	inter-class variance (between class variance)–first objective function.
$\sigma^2{}_i(T)$	variance of the i^{th} Gaussian curve.
$\omega(k)$	Zeroth-order cumulative moments of the histogram up to k^{th} gray level.

References

[1] R.D. Braatz, Advanced control of crystallization processes, *Annual Reviews in Control*, **26**(1), 87– 99 (2002).

[2] P.A. Larsen, D.B. Patience and J.B. Rawlings, Industrial crystallization process control, *Control Systems, IEEE*, **26**(4), 70– 80 (2006).

[3] J.D. Benoît Presles, Gilles Févotte and Jean-Charles Pinoli, Novel image analysis method for in situ monitoring the particle size distribution of batch crystallization processes, *Journal of Electronic Imaging*, **19**(03), (2010).

[4] Y. Zhou, R. Srinivasan and S. Lakshminarayanan, Critical evaluation of image processing approaches for real-time crystal size measurements, *Computers and Chemical Engineering*, **33**(5), 14 (2009).

[5] O. Monnier, G. Fevotte, C. Hoff and J.P. Klein, Model identification of batch cooling crystallizations through calorimetry and image analysis, *Chemical Engineering Science*, **52**(7), 1125– 1139 (1997).

[6] M. Oullion, F. Puel, G. Févotte, S. Righini and P. Carvin, Industrial batch crystallization of a plate-like organic product. In situ monitoring and 2D-CSD modelling: Part 1: Experimental study, *Chemical Engineering Science*, **62**(3), 820–832 (2007).

[7] C.J.G. Plummer and H.H. Kausch, Real-time image analysis and numerical simulation of isothermal spherulite nucleation and growth in polyoxymethylene, *Colloid and Polymer Science*, **273**(8), 719–732 (1995).

[8] M.-N. Pons and H. Vivier, Crystallization monitoring by quantitative image analysis, *Analytica Chimica Acta*, **238**(0), 243–249 (1990).

[9] F. Puel, G. Févotte and J.P. Klein, Simulation and analysis of industrial crystallization processes through multidimensional population balance equations. Part 2: a study of semi-batch crystallization, *Chemical Engineering Science*, **58**(16), 3729–3740 (2003).

[10] F. Puel, P. Marchal and J.P. Klein, Habit transient analysis in industrial crystallization using two dimensional crystal sizing technique, *Chemical Engineering Research and Design*, **75**(2), 193–205 (1997).

[11] P.A. Larsen, J.B. Rawlings and N.J. Ferrier, An algorithm for analyzing noisy, in situ images of high-aspect-ratio crystals to monitor particle size distribution, *Chemical Engineering Science*, **61**(16), 5236–5248 (2006).

[12] P.A. Larsen, J.B. Rawlings and N.J. Ferrier, Model-based object recognition to measure crystal size and shape distributions from in situ video images, *Chemical Engineering Science*, **62**(5), 1430–1441 (2007).

[13] Y. Zhou, X.-T. Doan and R. Srinivasan, Real-time imaging and product quality characterization for control of particulate processes, in W. Marquardt, C. Pantelides (Ed.), *Computer Aided Chemical Engineering*, Elsevier, Garmisch-Partenkirchen, pp. 775–780 (2006).

[14] J.M. Korath, A. Abbas and J.A. Romagnoli. Monitoring of Crystallization Processes: a Novel Approach for the Separation of Touching Edges in Crystal Particle Images. The Proceedings of AIChE Annual Meeting; San Francisco, 2006.

[15] J.M. Korath, A. Abbas and J.A. Romagnoli, Separating touching and overlapping objects in particle images—A combined approach, *Chemical Engineering Transactions*, **11**, 167–172 (2007).

[16] I.D. Mironescu and C. Mironescu, Image analysis for crystallization process control, *Journal of Agroalimentary Processes and Technologies*, **12**(1), 7–12 (2006).

[17] C.A. Glasbey, An analysis of histogram-based thresholding algorithms, *CVGIP: Graphical Models and Image Processing*, **55**(6), 532–537 (1993).

[18] P.K. Sahoo, S. Soltani, A.K.C. Wong and Y.C. Chen, A survey of thresholding techniques, *Computer Vision, Graphics, and Image Processing*, **41**(2), 233–260 (1988).

[19] K.R. Periasamy and S. Lakshminarayanan, Estimation of crystal size distribution based on two dimensional characteristics: an exploration using artificial images, *International Journal of Advances in Engineering Sciences and Applied Mathematics*, 1–13

[20] G. Bebis, The Geometry of Perspective Projection, 27/12/2011, www.cse.unr.edu/~bebis/CS791E/Notes/PerspectiveProjection.pdf (accessed December 12, 2012).

[21] P.A. Larsen and J.B. Rawlings, Assessing the reliability of particle number density measurements obtained by image analysis, *Particle and Particle Systems Characterization*, **25**(5–6), 420–433 (2008).

[22] Mathworks, R (2011b) Documentation - Image Processing Toolbox, 1/12/2011, www.mathworks.com/help/toolbox/images/ref/rgb2gray.html (accessed December 12, 2012).

[23] M.C. Gino, Noise, Noise, Noise, www.astrophys-assist.com/educate/noise/noise.htm (accessed December 12, 2012).

[24] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Pearson Education, Upper Saddle River, NJ (2008).

[25] R.C. Gonzalez, R.E. Woods and S.L. Eddins, *Digital Image Processing Using MATLAB*, Gatesmark Publishing, Singapore (2011).

[26] P. Thierry, A new method for grey-level picture thresholding using the entropy of the histogram, *Signal Processing*, **2**(3), 223–237 (1980).

[27] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Transactions on Systems, Man and Cybernetics*, **9**(1), 62–66 (1979).

[28] J. Kittler and J. Illingworth, Minimum error thresholding, *Pattern Recognition*, **19**(1), 41–47 (1986).

[29] G. Bebis, Thresholding, 27/12/2011, www.cse.unr.edu/∼bebis/CS791E/Notes/Thresholding.pdf (accessed December 12, 2012).

[30] M. Sezgin and B. Sankur, Survey over image thresholding techniques and quantitative performance evaluation, *Journal of Electronic Imaging*, **13**(1), 146–168 (2004).

[31] A. Nakib, H. Oulhadj and P. Siarry, Image histogram thresholding based on multiobjective optimization, *Signal Processing*, **87**(11), 2516–2534 (2007).

[32] A. Nakib, H. Oulhadj and P. Siarry, Image thresholding based on Pareto multiobjective optimization, *Engineering Applications of Artificial Intelligence*, **23**(3), 313–320 (2010).

[33] Z. Xinming and L. Chunhong, A two-dimensional image thresholding method based on multiobjective optimization, in *Proceedings of the International Symposium on Computer Network and Multimedia Technology*, Wuhan, China, pp. 1–5 (2009)

[34] G.P. Rangaiah (Ed.), *Multi-objective Optimization Techniques and Applications in Chemical Engineering*: World Scientific Publishing, Singapore (2009).

[35] H. Tamaki, H. Kita and S. Kobayashi, Multi-objective optimization by genetic algorithms: a review, in *Proceedings of the IEEE International Conference on Evolutionary Computation*, Nagoya, Japan, pp. 517–522 (1996).

[36] K. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, Boston (1999).

[37] K. Miettinen and J. Hakanen, Why use interactive multi-objective optimization in chemical process design? In G.P. Rangaiah (ed.), *Multi-Objective Optimization Techniques and Applications in Chemical Engineering*, World Scientific Publishing, Singapore, pp. 153–188 (2009).

[38] N. Srinivas and K. Deb, Multiobjective optimization using nondominated sorting in genetic algorithms, *Evolutionary Computation*, **2**, 221–248 (1994).

[39] K.A. Dowsland, Simulated annealing, in C.R. Reeves (ed.), *Modern Heuristic Techniques for Combinatorial Problems*, John Wiley & Sons, Inc., New York, pp. 20–69 (1993).

[40] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller, Equation of state calculations by fast computing machines, *The Journal of Chemical Physics*, **21**(6), 1087–1092 (1953).

[41] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, Optimization by simulated annealing, *Science*, **220**(4598), 671–680 (1983).

[42] V. Černý, Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, *Journal of Optimization Theory and Applications*, **45**(1), 41–51 (1985).

[43] E.M. Kasprzak and K.E. Lewis, Pareto analysis in multiobjective optimization using the collinearity theorem and scaling method, *Structural and Multidisciplinary Optimization*, **22**(3), 208–218 (2001).

[44] B. Bhanu, L. Sungkee and S. Das, Adaptive image segmentation using genetic and hybrid search methods, *IEEE Transactions on Aerospace and Electronic Systems*, **31**(4), 1268–1291 (1995).

[45] N. Instruments, Image Analysis and Processing, 1/12/2011. http://zone.ni.com/devzone/cda/tut/p/id/3470#toc2 (accessed December 12, 2012).

[46] G. Toussaint, Solving geometric problems with the rotating calipers, in *Proceedings of the IEEE Mediterranean Electrotechnical Conference,* Athens, Greece, pp. 1–8 (1983).