Team Number :	APMCM96330

Problem Chosen :	A

2019APMCM summary sheet

Study on the melting law of silicon dioxide based on the image processing technology

Abstract

This work is aimed at modeling a kind of silicon dioxide’ s melting law which can not be measured by direct contact means but can only be measured indirectly by image monitoring means. In this paper, the determination of silicon dioxide melting law is realized by image processing technology.

In problem 1, we establish an image segmentation model based on K-means algorithm, using this mature clustering algorithm to achieve the effective segmentation of the original image information, extracting the contour information of silicon dioxide from the raw image, at the same time using the Canny filtering technology to remove the noise around the image contour, using corrosion and expansion technology to fine the effective contour area The binary image with obvious contour feature of silica is obtained by adjusting. Furthermore, the two-dimensional coordinate system of the plane where the image is located is established, and the position of the center of mass of the silicon dioxide area in the binary image is determined by using the fast search algorithm. Finally, the coordinates of the center of mass of the silicon dioxide in 114 images are counted, and the relationship between the position of the center of mass and the discrete time point is obtained, and then the motion of the silicon dioxide center is satisfied by the high-powers function fitting function curve of the property. Through the error analysis of the residual function, the fitting results show that the equation of motion of the center of mass of silica is inclined to a 10-powers function.

In problem 2, we select the area index as the feature quantity in the process of silica melting, and only the cross-section image of crucible and the contour image of silica are retained in the image area by using image processing technology. Furthermore, the image processing function of MATLAB software is used to count all pixels surrounded by two kinds of contour lines. Considering that the ratio of the total number of pixels in two regions is the area ratio of the two regions, and the cross-sectional area of the crucible is a certain value, we associate the ratio of the total number of pixels in two regions with their area ratio, so as to determine the area of the silicon dioxide profile in unit time. Finally, the area data of silicon dioxide profiles in 114 drawings are counted, and the relationship between the area of silicon dioxide profiles and discrete time points is obtained. Further, the function fitting toolbox in MATLAB is used to fit the melting process of silicon dioxide, and the error analysis of the fitting results is carried out. The fitting results show that the melting law of silicon dioxide with time tends to a 3 powers function.

In problem 3, we start from a new idea and use the edge filtering technology to filter the original plane binary image twice. The purpose of this filtering is to mine the space area of the binary image that has not been detected in the past, to get the hidden gully area of the silicon dioxide stereo image surface, which indirectly plays the role of mining the third coordinate of the image's surface. After denoising and target extraction, a new binary two-dimensional image with the information of the third coordinate is obtained. Further, the volume of silicon dioxide in each image is estimated by the proportion between the number of pixels and the measurable amount, so that the relationship between the volume and the discrete time point is obtained. Further, the function fitting of the melting process of silica is carried out. The fitting results show that the melting law of silicon dioxide with time is more inclined to a kind of 2-powers function, while the melting rate and time tend to a linear relationship, which is basically consistent with the conclusions of some references.

Key words: Binaryzation; K-means algorithm; Image processing; Function fitting;.

Contents

I. Overview

1.1 Background........ 1
1.2 Restatement of the Problems........ 1
1.3 Analysis of the Problems........ 2
1.3.1 Analysis of the Problem1........ 2
1.3.2 Analysis of the Problem2........ 2
1.3.3 Analysis of the Problem3........ 3
1.4 General Assumptions........ 3
1.5 Symbol Description........ 4

II. Model Establishment and Solution of Problem 1........ 5

2.1 Problem 1 Solving Process........ 5
2.2 Feature Extraction from Raw Data in the 114 images........ 5
2.2.1 Gray-Scale Image Transformation and its Enhancement........ 6
2.2.2 The K-means clustering algorithm........ 7
2.2.3 Smoothing processing after the K-means algorithm........ 8
2.3 Determine the coordinates of the silicon dioxides centroid........ 10
2.3.1 Establish a coordinate system of whole image area........ 10
2.3.2 Calculate the centroid position........ 11
2.3.3 Determining the track of the centroid........ 12

III. Model Establishment and Solution of Problem 2........ 15

3.1 Problem 2 Solving Process........ 15
3.2 Defining the feature of silicon dioxide’s edge profile feature........ 15
3.3 Calculate the area of silicon dioxide’s image contour........ 16
3.3.1 Relationship between the silicon dioxide’s contour area and the crucible's cross section area........ 16
3.3.2 Obtain the area data sets corresponding to the time sequence........ 17
3.4 Drawing the melting curve of silicon dioxide........ 18

IV. Comparative Analysis and Solution of Problem 3

4.1 The 3D reconstruction model of the 2D image........ 20
4.2 The law of mass loss in the silicon dioxide's melting process........ 23
4.3 The actual melting rate of the silicon dioxide........ 24

V. Strengths and Weaknesses

5.1 Strengths........ 26
5.2 Weaknesses........ 26

References

Appendix

I. Overview

1.1 Background

Iron tailings are the waste after ore dressing, the main ingredient of Iron tailings is silicon dioxide. Nowadays, the comprehensive utilization rate of tailings in China is only 7%. Therefore, the comprehensive recovery and utilization of iron tailings has been widely concerned by the whole society.

For a long time, many researchers have been studying the melting law of iron tailings ore at high temperature to provide guidance for the tailing addition and heat compensation in the process of slag cotton preparation, thus indirectly improve the direct fiber forming technology of blast furnace slag, then turn waste into treasure.After years of research, experts and scholars at home and abroad have made a lot of research results. Especially the application of CCD image processing technology directly solves the problem that routine detection equipment is inconvenient to use in high temperature environment,it also greatly facilitate the appearance detection of iron tailings in the melting process.

For all kinds of objects that are inconvenient to be directly measured, the image is first magnified by means of microscope imaging, then the image is collected by CCD camera, and finally the image is processed by computer to obtain the result. The technology has been widely used in aerial remote sensing measurement, microdimension measurement and appearance detection of precision and complex parts.

1.2 Restatement of the Problems

In order to improve the direct fiber forming technology of blast furnace slag, it is necessary to reveal the melting law of iron tailings in blast furnace slag. Researches haven show that the melting behavior of iron tailing can be represented by the melting behavior of silicon dioxide, therefore, the melting process of SiO2 particles at high temperature can be studied to represent the melting of iron tailings. To solve the problem that the service life of routine detection equipment is very short under the high-temperature environment above 1500℃ in the molten pool during the process of observation, as shown in the figure 1, the research team used a kind of rifted CCD video recording system with amplification effect to obtain dynamic visual data of silicon dioxide in the high-temperature molten pool (sequence image under time sequence), and also has observed the real-time melting rate of silicon dioxide in the time sequence through video analysis.

It is required to conduct the following analysis on the time sequential image of silicon dioxide in high-temperature molten pool during the melting process in the Attachment(114 images in total; the serial number of document name is the time sequence, collecting one image every other 1s)

Problem 1: During the melting process, the position of silica particles in the high-temperature molten pool is changing constantly. Please establish a mathematical model to track the centroid position of silicon dioxide particles during the melting process, and present the motion trail of centroid of silicon dioxide.

Problem 2: Participants are invited to select and establish indicators (such as shape, perimeter, area, generalized radius, etc.) representing the edge contour characteristics during the melting process of silicon to represent the melting process of silica.

Problem 3: The key parameter for the corrosion resistance of silica is mass, which is proportional to the cubic volume. Please estimate the actual melting rate of silicon dioxide according to the characteristic index of the edge contour of silicon dioxide in the second question.

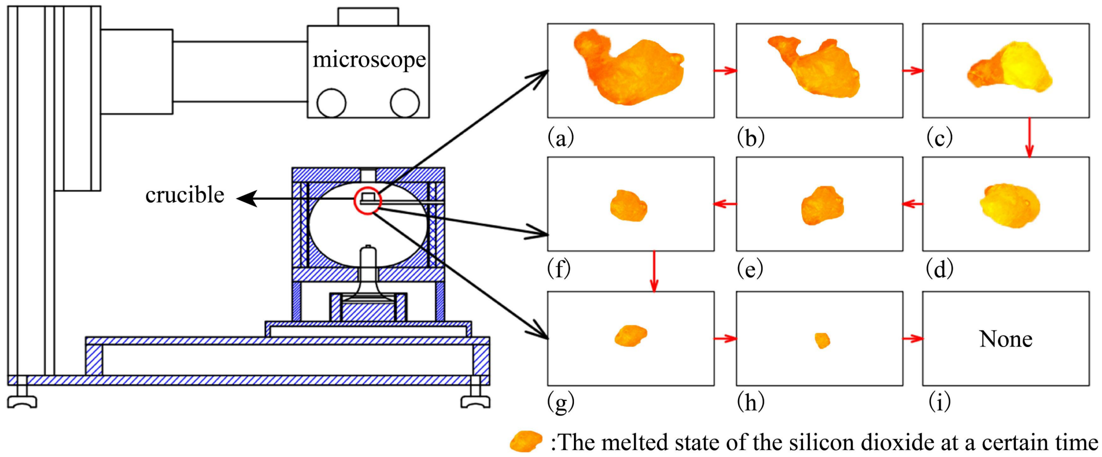


Figure 1. The research device of SiO2 melting law and its principle.

1.3 Analysis of the Problems

1.3.1Analysis of the Problem1

For problem 1, so as to establish a centroid's displacement and movement track model of the silicon dioxide in its melting process, firstly we should determine the coordinate of the silicon dioxide's centroid in the image plane (2D) within one second based on the 114 sequential melting images of silicon dioxide obtained by CCD video shooting system. Because the position of the images's whole plane has been fixed, so if the position of the silicon dioxide's centroid in the whole image plane can be determined per second, it's means that the distribution characteristic of the centroid's position in the 114 seconds can be further determined. Thus, we can get the relations between the position of the centroid and the discrete time points, then we can further get the function curve which could satisfy the motion characteristics of silicon dioxide's centroid through an appropriate numerical fitting.

1.3.2Analysis of the Problem2

For problem 2, we analyze that, the first feature of silicon dioxide's contour acquired by CCD camera is its contour area, so we should choose the area factor as the index to describe the edge contour feature in the silicon dioxide's melting process.

Firstly, we should use the image processing technology to remove the black background of the whole image, at this time, the image area only retains the image of crucible section and the contour image of silicon dioxide. Then, we should use the image processing function in MATLAB to count all the pixels surrounded by the two kinds of contour lines. It can be considered that the ratio of the two pixels is the area ratio of the two, hence we can further associate the ratio of the two pixels with the area ratio, hence, the area of the silicon dioxide's profile can be determined per second. According to the extracted area information, we should also use the function fitting toolbox in MATLAB to fit the silicon dioxide's melting process.

1.3.3Analysis of the Problem3

For problem 3, accroding to the problem 3's information, it can be kown that the essential parameter to express the melting rate of silicon dioxide is its mass rather than its two-dimensional area, so it is necessary to find a relationship between the mass and time in the melting process of silica, and because the volume is a 3D parameter, so it's necessary to reconstruct the raw 2D image. In generally, the 3D reconstruction process of 2D image refers to the operation of reconstructing the 3D shape of the surface of 2D gray-scale image, and the algorithms used in this reconstruction process are usually is the SFS and SFX, but after our analysis, these algorithms have high requirements for the foreground-background gray-scale differences、 image luminosity and resolution, that is, they need to gray-scale image with high resolution and big difference from background color. However, most of the 114 images provided by the data can not meet this kind of requirements. Even if the image enhancement preprocessing operation is carried out, it can not guarantee that the difference of foreground and background gray of most of the images can meet the requirements of 3D reconstruction. So, we should relize the 3D feature information extraction in 2D image by the image processing techniques, we should try to mine the undetected space area of 2D binary image

1.4 General Assumptions

①It is assumed that the illumination factor of imaging environment is stable in each time interval and the position of light source is fixed;
②Ignore a small part of silicon dioxide's separation because of the high temperature in the melting process, only study the main part of silicon dioxide;
③Suppose the cross section of the crucible is a standard circle;
④It is assumed that the density of silicon dioxide is uniformly distributed;
⑤Since the mass of silicon dioxide is directly proportional to its volume, we assume that the density of silicon dioxide particles is unit 1, and further change the relationship between mass and time in the melting process into the relationship between volume and time.

1.5 Symbol Description

We define the following variables of the whole model in this table.

Symbols	**Definitions**	**Symbols**	**Definitions**
$\beta_{\mu\nu}$	Fuzzy field's matrix	H	The space height
$g_{\mu\nu}$	Gray value	S_g	Bottom area of the gray
g_m	Maximum gray value	a	The numerical value of the long axis of ellipsoid
λ_a	Transformation coefficient	b	The numerical value of the short axis of ellipsoid
λ_b	Transformation coefficient	c	The distance between the focus and origin
$d(x,y)$	Euclidean distance	V_S	actual volume of silicon dioxide
Θ	Dilation operation	V_C	The effective volume of silicon dioxide
$\oplus$	Erosion operation	N	Pixel's number
$R(x,y)$	Image's coordinate	ρ_m	the density of silicon dioxide
X	The coordinate x	M	The mass of the silicon dioxide
Y	The coordinate y	$M(t)$	The mass function
S_c	The cross-section area of the crucible	$v(t)$	Melting function
S_d	The area of the SiO_2's edge contour	$*$	Convolution operation
R	The radius of the crucible	h	The height of th gully

II. Model Establishment and Solution of Problem 1

2.1 The Solving Process of Problem 1

In order to establish the centroid's displacement and movement track model of the silicon dioxide in the melting process, we think that we should firstly determine the coordinate of the silicon dioxide's centroid in the image plane (2D) within one second based on the 114 sequential melting images of silicon dioxide obtained by CCD video shooting system. Because the position of the images's plane has been fixed（the reason is that the position of CCD video shooting system is fixed）, so if the position of the silicon dioxide's centroid in the image plane can be determined every 1 second, it's means that the distribution of the centroid's position in the 114 seconds can be further determined. Hence, we can get the relations between the position of the centroid and the discrete time points, then we can further get the function curve which could satisfy the motion characteristics of silicon dioxide's centroid through an appropriate numerical fitting. The model's establishment process and logical relationship of problem 1 are shown in Figure 2.

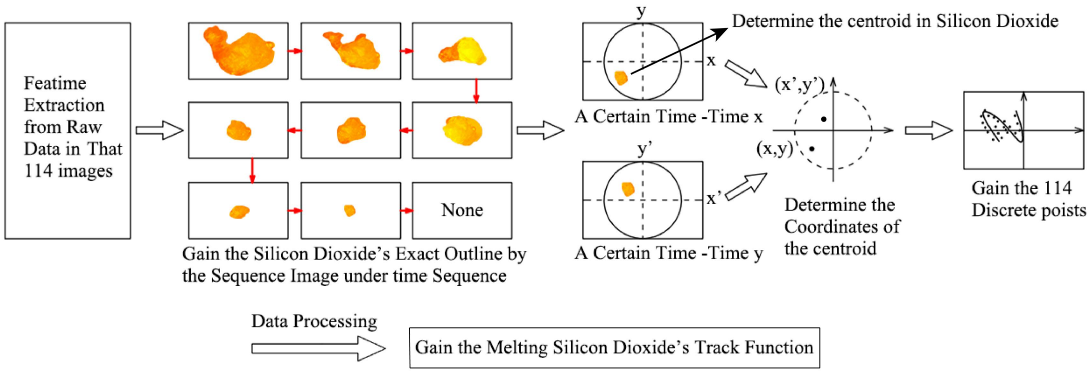


Figure 2. The establishment process and its logical of problem 1.

2.2 Feature Extraction from Raw Data in the 114 images

Now we need to extract the required characteristic information from the 114 raw images, we should get information about the location and outline of silicon dioxide in these raw images, because we need to further determine the exact coordinate of the centroid by determining the location and its outline of silicon dioxide in these images. It is necessary to use a images segmentation technology to extract the location and outline information of silicon dioxide from the raw images, the segmentation technology refers to a kind of image processing technology that separates the object or region in the original image from the complex background[1], and the algorithm to realize this technology is mainly clustering algorithm, and this kind of algorithm uses a series of feature points to represent the pixel points in the 2D space of these images[2], then these points can be segmented effectively according to the aggregation degree of feature points in feature space. Finally these points are mapped back to the space of the raw images, so as to realize a image segmentation at once.

In order to ensure the accuracy of image segmentation, and realize the fast and correct segmentation process, we use a mature hierarchical clustering analysis method —K-means clustering algorithm, it is the core image segmentation algorithm in this model. In the specific operation, we established the image segmentation model based on the K-means algorithm as shown below, and tried to use MATLAB software to segment the contour pattern of silicon dioxide from the raw sequential images. The whole image processing flow is shown in Figure 3.

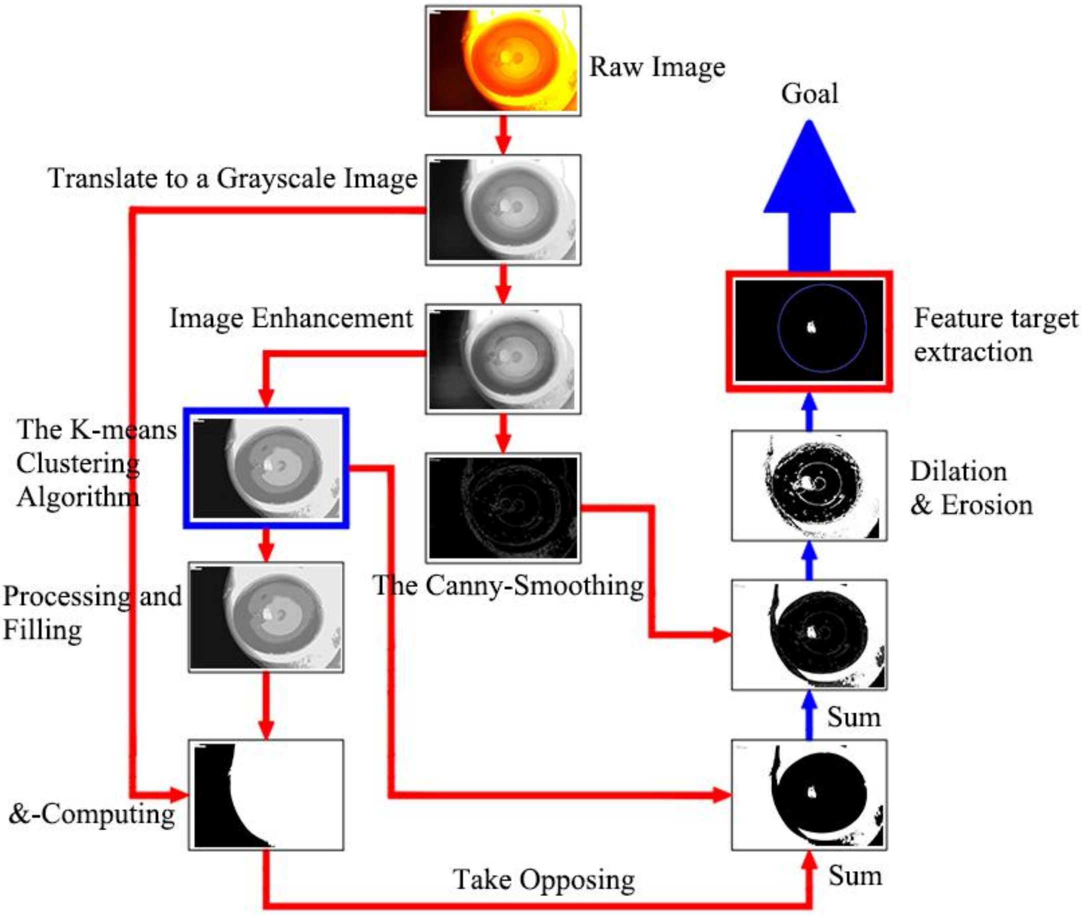


Figure 3. The image processing flow to feature extraction.

2.2.1 Gray-Scale Image Transformation and its Enhancement

First of all, it's necessary to transform the raw image into a gray-scale image, we use the binary method to make the image become a classical gray-scale image, and the pixel points in the gray-scale image have appeared dispersedly. However, due to a series of gray-scale images（for example, the gray-scale image from 540 seconds to 544 seconds） have a low contrast, that is the difference between the silicon dioxide's image and the background image is low, hence it's necessary to strengthen the image of these low contrast images before hierarchical clustering analysis to highlight the difference between the silicon dioxide's image and the background image. In this regard, we use the image enhancement method based on the fuzzy set to extract the fuzzy features of the gray-scale images, and establish a mathematical model for the transformation from the spatial domain to the fuzzy domain, which is expressed as[3]

$$\beta_{\mu v}=G\left(g_{\mu v}\right)=\left[1+\frac{g_m-g_{\mu v}}{\lambda_a}\right]^{-\lambda_b} \tag{1}$$

it also can be expressed in matrix form as

$$\begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1v} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2v} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{\mu 1} & \beta_{\mu 2} & \cdots & \beta_{\mu v} \end{bmatrix} = \left\{1+\frac{1}{\lambda_a}\begin{bmatrix} g_{m1} & 0 & \cdots & 0 \\ 0 & g_{m2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g_{mn} \end{bmatrix} - \frac{1}{\lambda_a}\begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1v} \\ g_{21} & g_{22} & \cdots & g_{2v} \\ \vdots & \vdots & \ddots & \vdots \\ g_{\mu 1} & g_{\mu 2} & \cdots & g_{\mu v} \end{bmatrix}\right\}^{-\lambda_b}$$

where $\beta_{\mu v}$ is the matrix element of fuzzy field, $g_{\mu v}$ is the gray value of the current pixel in the gray-scale image, g_m is the current maximum gray value of gray-scale image, and λ_a 、 λ_b is the transformation coefficient. Then we need to modify the membership of $\beta_{\mu v}$, if $\beta_{\mu v}$ within $[0,1/2]$, the membership degree based on fuzzy enhancement factor is

$$INT\left(\beta_{\mu v}\right)=2\cdot\left[\beta_{\mu v}\right]^2 \tag{2}$$

and if $\beta_{\mu v}$ within $[1/2,1]$, the membership degree based on fuzzy enhancement factor is

$$INT\left(\beta_{\mu v}\right)=1-2\cdot\left[1-\beta_{\mu v}\right]^2 \tag{3}$$

select the range of β that can enhance the image, that is $\beta_{\mu v}$ with $[1/2,1]$, can be enhancing the image which in fuzzy domain. At last, we need to find the fuzzy matrix's inverse matrix, it will be expressed as[3]

$$g'_{\mu v}=G^{-1}\left(\beta'_{\mu v}\right) \tag{4}$$

the enhancement effect is shown in Figure 4.

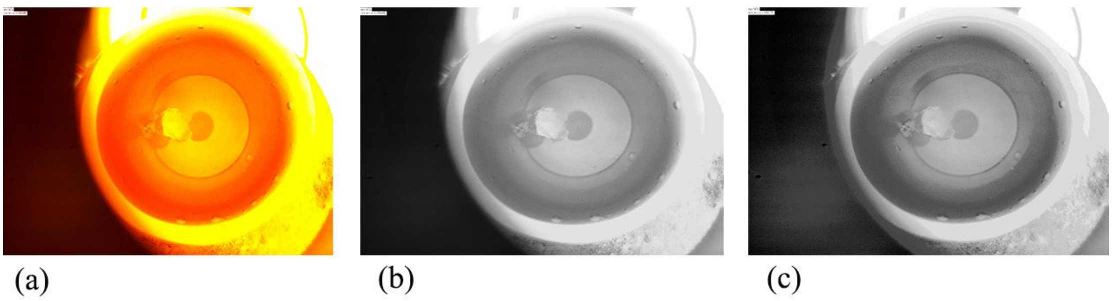


Figure 4. Gray-scale image Transformation and its Enhancement. (a):The raw image; (b):The gray-scale image; (c):Gray-scale image's enhancement.

2.2.2 The K-means clustering algorithm

After the image enhancement, we sample these plane pixels to form a set of feature points, called them data set, which can be expressed as

$$Y_\mu=\{y_m \mid m=1,2,3,...,et\ al\}=\{y_1+y_2+y_3+...\} \tag{5}$$

where, the data type of $\boldsymbol{K}$ vector is represented by d spatial dimensions $\boldsymbol{K}_1$, $\boldsymbol{K}_2$, $\boldsymbol{K}_3,...,\boldsymbol{K}_d$. Generally, two d-dimensional row vectors $\boldsymbol{y}_i$、$\boldsymbol{y}_j$ are selected as data sample points in the comparison of correlation (similarity), which are expressed as

$$y_i=\left[y_{i1},y_{12},y_{13},...,y_{id}\right] \tag{6}$$

$$y_j=\left[y_{j1},y_{j2},y_{j3},...,y_{jd}\right] \tag{7}$$

where, each element in $\boldsymbol{y}_i$ (d in total) and each element in $\boldsymbol{y}_j$ (d in total) are the values of $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ in d spatial dimensions respectively, and the corresponding difference between the two is a reflection of clustering correlation degree, that is

$$\delta_{ij}=\left|x_{ik}-x_{jk}\right|,\ k\in\left[1,d\right] \tag{8}$$

It can be proved that the larger the d value is, the smaller the correlation and similarity of the two point sets are[4], and the more impossible to cluster, so in this case, the two pixel points will not gather in the same color block. The k-means algorithm divides the existing data into many different categories through several iterations, generally into two sample types, and can gradually update the cluster center many times to achieve the desired goal. According to the k-means algorithm, in order to compare the correlation between two pixel samples, we need to use the Euclidean distance between them, it can be expressed as[3,4]

$$d\left(y_i,y_j\right)=\sqrt{\sum_{k=1}^{d}\left(y_{ik}-y_{jk}\right)^2},\ k\in\left[1,d\right] \tag{9}$$

so far, the similarity evaluation between sample points is over.

After determining the similarity evaluation criteria, it is necessary to define the clustering performance of each pixel sample. Different clustering performance leads to different pixels gathering to different color layers. The clustering performance can be described by a kind of clustering performance criterion function[3]. In order to describe the clustering performance of each pixel in the gray-scale image, we choose the error square sum function as the evaluation standard of the clustering ability between pixels, and define the evaluation standard as B, which can be shown as

$$B=\sum_{i=1}^{k}\sum_{p\in Y_\mu}\left\|p-Z_i\right\|^2,\ k\in\left[1,d\right],\ p\in Y_\mu \tag{10}$$

where, Y_i represents k cluster subsets; Y refers to the data set defined in 1; Z_i refers to the cluster center of cluster subsets; B is the sum of mean square deviation; p refers to any point in 2D space. Through B function, we can judge the clustering performance of each pixel on the basis of MATLAB software, and further form regular clustering color layer through pixel clustering. The clustering graph obtained from the gray-scale graph is shown in Figure 5.

Finally, we calculate the data similarity in the sample cluster. We know that the data sample has k cluster subsets. In the process of K-means algorithm, we need to randomly assign the data object to k sample clusters. The value of k is controlled by us, and in this model, we set k = 2. After assigning data objects to each cluster, we need to calculate the average value of each cluster, which is the characteristic value of the cluster. Using MATLAB software, the data can be allocated to the next cluster independently. The flow chart of the algorithm is shown in Figure 6.

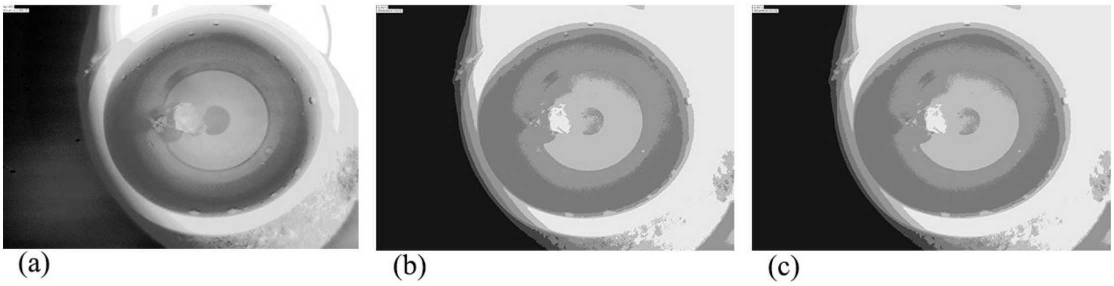


Figure 5. Clustered result of gray-scale images by using k-means algorithm. (a):The gray-scale image(Enhancement); (b):Clustered results of K-means; (c):Filling result.

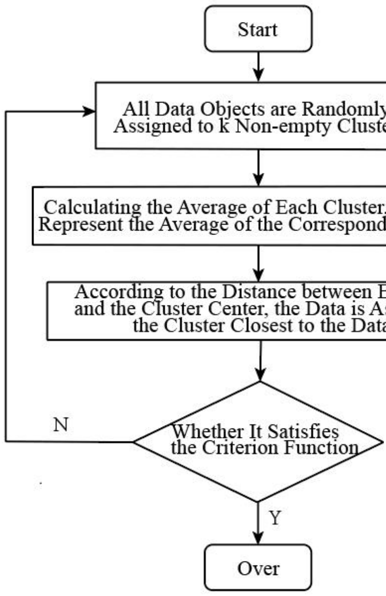


Figure 6. The calculate of the data similarity in the sample cluster.

2.2.3 Smoothing processing after the K-means algorithm

In the process of applying the model, it is found that there are still some interference factors such as burr, noise and so on in the contour of partial silicon dioxide image after image segmentation, which makes us difficult to extract the effective information of the image. In order to solve this kind of problem, we use canny filter function to eliminate the noise outside the contour, and use corrosion operation to basically eliminate the burr around the contour; for the modification of the shadow part of silicon dioxide's image, this paper uses the expansion theory to effectively fill the shadow part, has improved the accuracy of the effective area, the effect is shown in Figure 7. This kind of image processing mode is also a morphological model. To enhance silicon dioxide shadow as for the integrity of the image, we carries out morphological open operation and closed operation on the later image, and records that the elements of two different structures are C and D, then the open operation of D to C between the two structures is defined as

$$C \circ D = \left(C \Theta D\right) \oplus D, \quad C \Theta D = \left\{x_i, D_x \in C\right\} \qquad (11)$$

where, the Θ is dilation operation and the $\oplus$ is erosion operation[3].

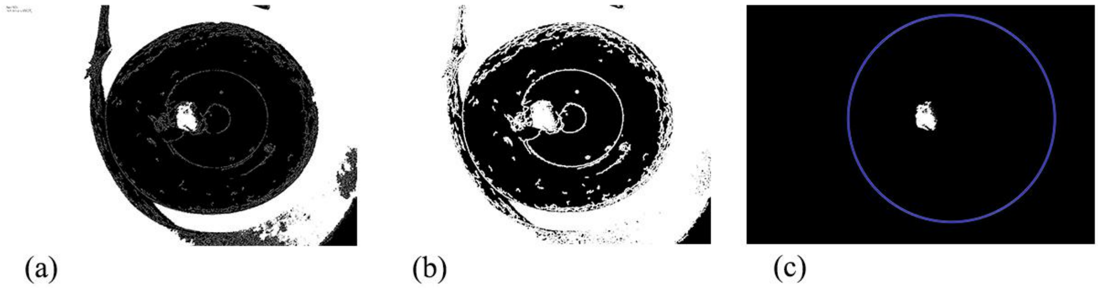


Figure 7. The segmentation image after dilation and erosion.

According to the above model, with the help of MATLAB software, we have segmented the effective area of 114 raw images. We have mined out the clear outline of the silicon dioxide image needed from the raw image, as shown in Figure 8. We have selected 6 processed images and shown them as follows, and we can see that the bright white area is the silicon dioxide obtained through image processing Contour information, through which we can further determine the position of the silicon dioxide's centroid per second.

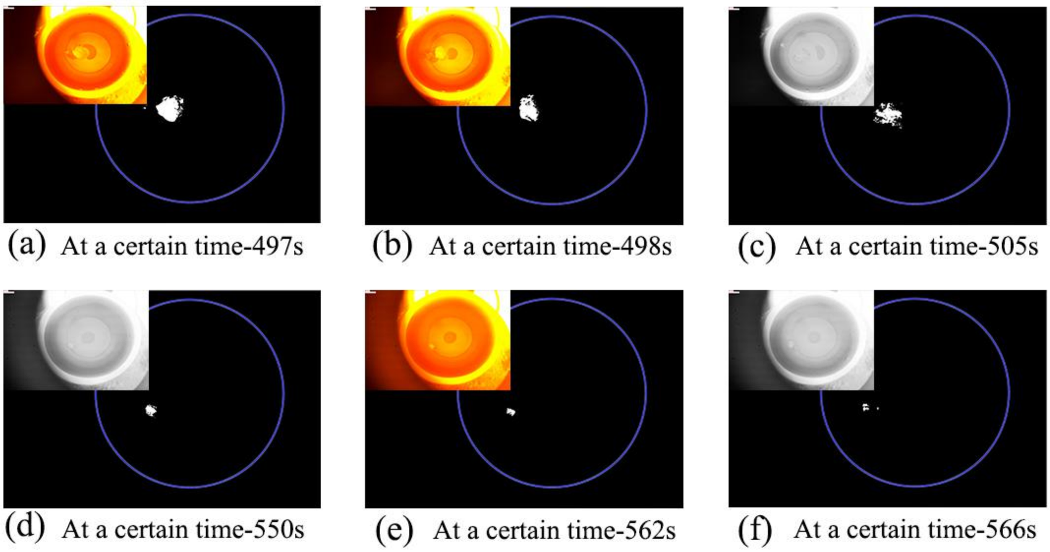


Figure 8. The silicon dioxide's outline characters (bright white area)

2.3 Determine the coordinates of the silicon dioxide's centroid

2.3.1 Establish a coordinate system of whole image area

The premise of determining the precise position and motion track of the silicon dioxide's centorid is to establish a suitable coordinate system in 2D plane, so as to determine the coordinate value of the silicon dioxide's centorid . Because the specifications of the images processed by this model are 1231 × 1792 (M × N, M = 1231, N= 1792), so we firstly divide the image into M × N pixel blocks, and specify the exact position of each pixel block in 2D space Set and sequence values. The establishment of a two-dimensional rectangular coordinate system is shown in Figure 9, in which each pixel represents a coordinate (x, y). In the later stage, when we need to determine the position of the silicon dioxide's centorid and tracking the trajectory of the silicon dioxide's centorid, we can record the coordinates of the center of mass at any time.

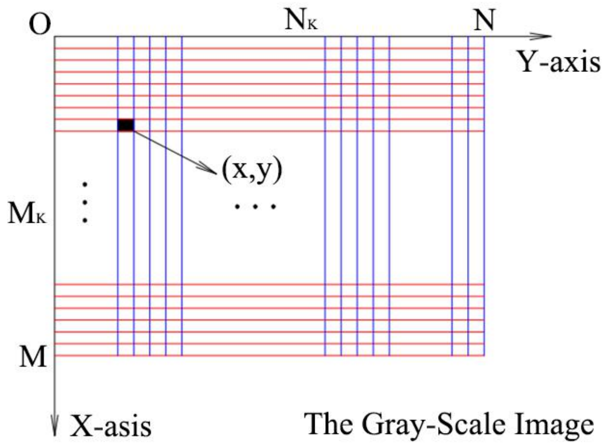


Figure 9. The coordinate system of whole image area(in gray-scale image)

2.3.2 Calculate the centroid position

As shown in Figure 9，the feature contour of the silicon dioxide image which we extracted is in a strict binary image. We define the binary image as the binary image *R* (x, y) of M × N dimension, where the silicon dioxide's image contour we studied is defined as the target area, so it is defined as area A (bright white area as shown in Figure 8), and the remaining background color (the black area as shown in Figure 8 Domain) as area B, so the area A and B an be expressed as[5]

$$R(x,y)=\begin{cases}0, & (x,y)\in A\\ 1, & (x,y)\in A\end{cases} \qquad (12)$$

$$R'(x,y)=\begin{cases}\varnothing, & (x,y)\in B\\ \varnothing, & (x,y)\in B\end{cases} \qquad (13)$$

where, the $\varnothing$ refer to the empty sets.

And our purpose is to determine the geometric center of the 2D geometry surrounded by this outline, that is the center of mass position of silica. Let the centroid coordinate of the target area be R_0 (x_0, y_0), and define its coordinate value as

$$x_0=\frac{1}{\sum_{(x,y)\in A}^{A} R(x,y)}\cdot\sum_{(x,y)\in A}^{A} x\cdot R(x,y) \qquad (14)$$

$$y_0 = \frac{1}{\sum\limits_{(x,y)\in A}^{A} R(x,y)} \cdot \sum_{(x,y)\in A}^{A} y \cdot R(x,y) \tag{15}$$

From the definition of formula (14) and formula (15), the value of x_0 and y_0 is essentially the mean value of the points set formed by the horizontal and vertical coordinates of the target area A.

In reference [6], the author points out an important property about the centroid of the target image, that is, the squares sum of the abscissa of the centroid of the target area A to the abscissa of all other points in the area A is the minimum compared with any other point. If the position of any point in a region is represented by *R*(i, j), then the convolution integral for searching the shortest distance between two points is defined as follows [6]

$$F_x(k) = \sum_{i=1}^{m}\sum_{j=1}^{n}(k-x_i)^2 \cdot R(i,j) = d(k,x_i) * R(x,y) \tag{16}$$

similarly, the convolution integral of y axis is expressed as

$$F_y(k) = \sum_{i=1}^{m}\sum_{j=1}^{n}(l-y_i)^2 \cdot R(i,j) = d(l,y_i) * R(x,y) \tag{17}$$

According to formula (16) and formula (17), when k=x_0 and l= y_0, two convolution integrals have minimum values. According to this, we use MATLAB software to continuously search a point wich has the smallest sum of distance to all points in area A, and eliminate the point with large deviation by overlapping technology, and finally get the accurate coordinate value of the center of mass. We calculated the centroid position of each image in 114 seconds, and randomly displayed 20 sets of coordinate data in tabular form as follows table 1. These discrete points are related to time.

Tabel 1. Centroid's coordinate data based on the Figure 9's theory.

Sequence /s	centroid's x-coordinate	centroid's y-coordinate	Sequence /s	centroid's x-coordinate	centroid's y-coordinate
497	578.24	906.86	543	646.02	822.02
498	580.39	872.56	544	650.02	806.02
499	603.51	874.49	545	673.02	793.02
510	601.02	887.02	558	673.02	758.02
511	601.02	888.02	559	695.02	769.02
512	599.02	886.02	560	688.02	774.02
524	600.02	863.02	578	712.02	895.02
525	603.02	856.02	579	710.02	895.02
526	610.02	855.02	580	714.02	895.02
530	608.02	852.02	600	452.02	962.02

2.3.3 Determining the track of the centroid

According to the 114 discrete points (the coordinate value of the centroid corresponding to the discrete time point), the trajectory function of the centroid can be established. Firstly, we express these discrete points in Origin 2017, and then use the function fitting toolbox in MATLAB to fit these discrete points with high-power function, the trajectory image obtained is shown in Figure 10. It should be noted that, because the relationship between the coordinate values of centroid's X、Y data and the time sequence cannot be displayed intuitively in the plane coordinate system. and even if the relationship between the coordinate values of X、Y data and the time sequence is expressed in the three-dimensional space, the trajectories formed have no practical significance. So we separate the data of X coordinate and Y coordinate, and respectively fit the relationship between the X、Y coordinate of the centroid and their corresponding time sequence, and the function relationship between the X、Y coordinate and time of the centroid is obtained respectively, hence, we obtain the motion track of the centroid's X direction and Y direction, and the tracking of the motion of the centroid is realized.

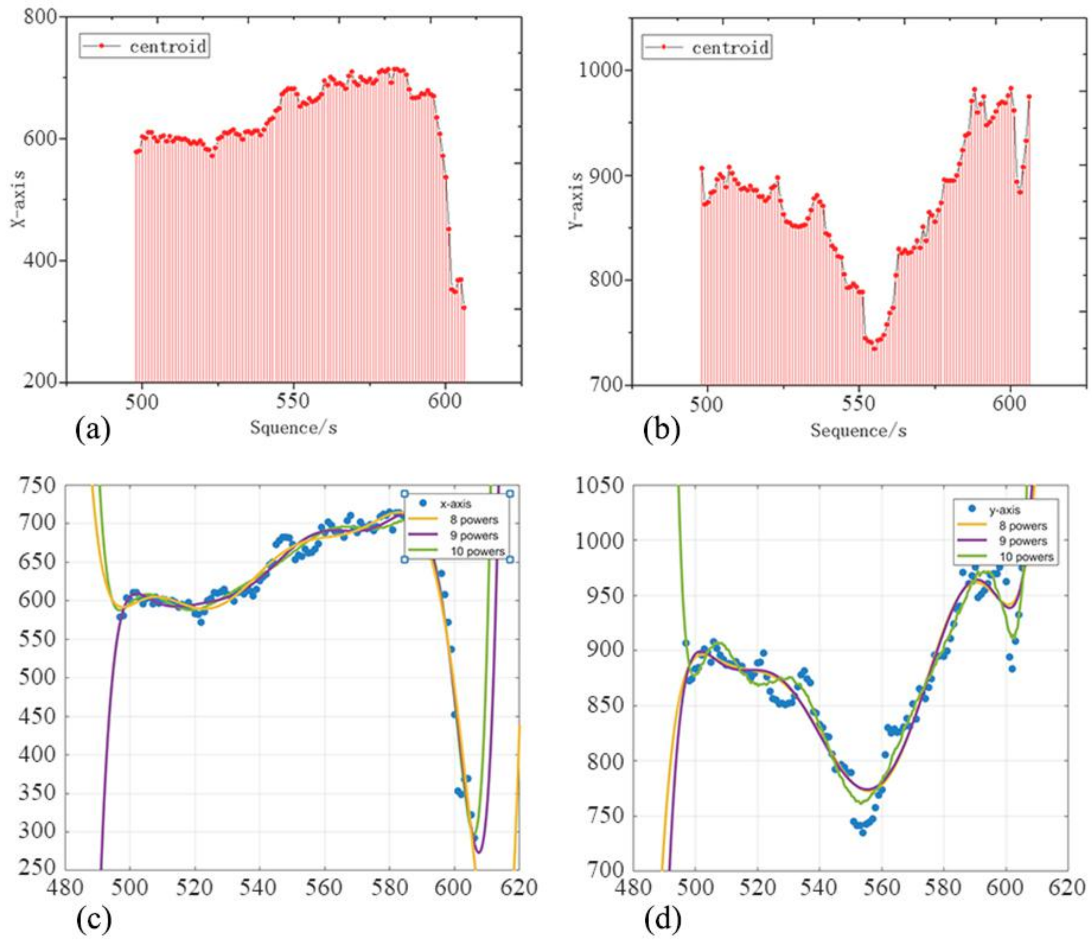


Figure 10. The spatial distribution of the centroid and its track function.
(a) and(b):The spatial distribution of the centroid's motion data;
(b) and(d) :The function relationship between the X、Y coordinate and time of

the centroid by the MATLAB's function fitting toolbox's analysis

The function expression of the centroid's motion track in the X direction obtained from the function fitting toolbox of MATLAB is respectively expressed as (three different fitting methods):

①The 8 powers function fitting results:

$$x = 1.5\times10^{-11}t^8 - 6.7\times10^{-8}t^7 + 0.00013t^6 - 0.14t^5 + 97t^4 - 4.2\times10^4t^3 + 1.2\times10^7t^2 - 1.8\times10^9t + 1.2\times10^{11}$$

②The 9 powers function fitting results:

$$x = 6.4\times10^{-13}t^9 - 3.2\times10^{-9}t^8 + 6.9\times10^{-6}t^7 - 0.0088t^6 + 7.3t^5 - 4\times10^3t^4 + 1.5\times10^6t^3 - 3.4\times10^8t^2 + 4.7\times10^{10}t - 2.8\times10^{12}$$

③The 10 powers function fitting results:

$$x = 1.7\times10^{-14}t^{10} - 9.5\times10^{-11}t^9 + 2.3\times10^{-7}t^8 - 0.00034t^7 + 0.33t^6 - 2.1\times10^2t^5 + 9.8\times10^4t^4 - 3\times10^7t^3 + 6.2\times10^9t^2 - 7.6\times10^{11}t + 4.1\times10^{13}$$

According to the residual function diagram corresponding to Figure 10(c) (as shown in Figure 11), it can be found that the fitting error of the 8 powers function is the smallest, so the expression of the 8 powers function can be used to represent the motion of the center of mass in the X direction.

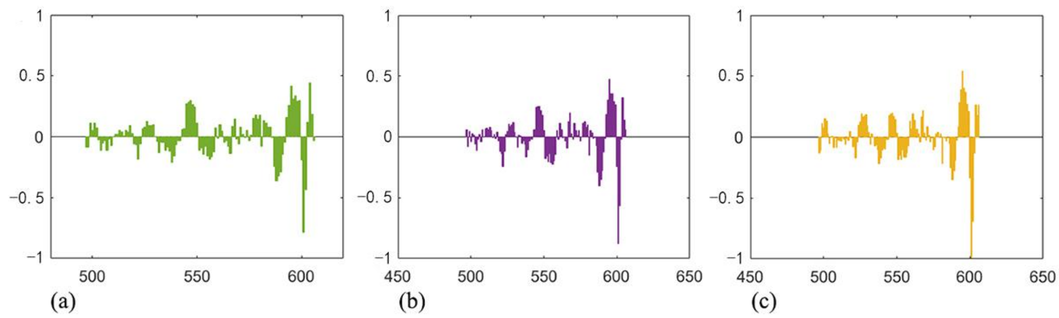


Figure 11. The residual graph of fitting function in X direction. (a):Residual graph corresponding to the 8 powers function; (b):Residual graph corresponding to the 9 powers function; (c):Residual graph corresponding to the 10 powers function.

Similarly, the centroid's motion track in the Y direction obtained from the function fitting toolbox of MATLAB is respectively expressed as:

①The 8 powers function fitting results:

$$y = -1.1\times10^{-12}t^8 + 5.6\times10^{-9}t^7 - 1.2\times10^{-5}t^6 + 0.015t^5 - 11t^4 + 5.3\times10^3t^3 - 1.6\times10^6t^2 + 2.6\times10^8t - 1.9\times10^{10}$$

②The 9 powers function fitting results:

$$y = 1.7\times10^{-13}t^9 - 8.6\times10^{-10}t^8 + 1.9\times10^{-6}t^7 - 0.0024t^6 + 2t^5 - 1.1\times10^3t^4 + 4.1\times10^5t^3 - 9.7\times10^7t^2 + 1.3\times10^{10}t - 8.2\times10^{11}$$

③The 10 powers function fitting results:

$$y = 3.9\times10^{-14}t^{10} - 2.1\times10^{-10}t^{9} + 5.3\times10^{-7}t^{8} - 0.00077t^{7} + 0.75t^{6} - 4.9\times10^{2}t^{5} + 2.3\times10^{5}t^{4} - 7.1\times10^{7}t^{3} + 1.5\times10^{10}t^{2} - 1.8\times10^{12}t + 9.7\times10^{13}$$

According to the residual function diagram corresponding to Figure 10(d) (as shown in Figure 12), it can be found that the fitting error of the 10 powers function is the smallest, so the expression of the 10 powers function can be used to represent the motion of the center of mass in the Y direction.

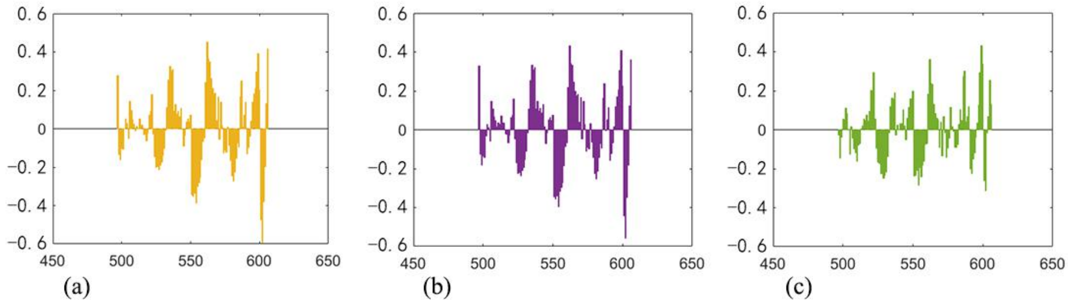


Figure 12. The residual graph of fitting function in Y direction. (a):Residual graph corresponding to the 8 powers function; (b):Residual graph corresponding to the 9 powers function; (c):Residual graph corresponding to the 10 powers function.

III. Model Establishment and Solution of Problem 2

3.1 The Solving Process of Problem 2

Because the first feature of silicon dioxide’s contour acquired by CCD camera is its contour area, so we choose the area factor as the index to describe the edge contour feature in the silicon dioxide’s melting process. Firstly, we use the image processing technology to remove the black background of the whole image, at this time, the image area only retains the image of crucible section and the contour image of silicon dioxide. Secondly, we have used the image processing function in MATLAB software to count all the pixels surrounded by the two kinds of contour lines. Considering that the ratio of the two pixels is the area ratio of the two, we can further associate the ratio of the two pixels with the area ratio, hence, the area of the silicon dioxide’s profile can be determined per second. According to the extracted area information, we have used the function fitting toolbox in MATLAB to fit the silicon dioxide’s melting process, and also have analyzed the error of the fitting results.

3.2 Defining the feature of silicon dioxide’s edge profile feature

Because the lens of CCD camera is vertically aligned with the crucible’s section, the first feature (also is the main feature) of the silicon dioxide contour collected by the camera is its contour area. Hence, we choose the area factor as the index to describe the edge contour feature in the process of silicon dioxide’s melting. The principle of characterizing the melting process of silicon dioxide with the contour area is to collect the data of the contour area of silicon dioxide at each time and form a complete data set. These data sets are discrete points related to the time sequence. According to these discrete points, the discrete graph between the contour area product of silicon dioxide and the time sequence can be determined, and then the method of function fitting can be used to find the one that meets the melting process Law.

The method of collecting silicon dioxide's profile area data should be based on the segmented effective image obtained in problem 1. The specific feature extraction idea is shown in Figure 13.

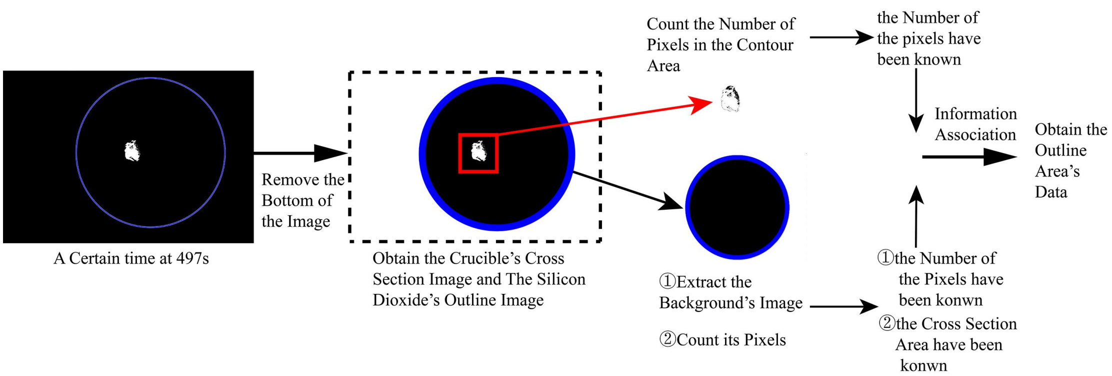


Figure 13. The extraction process of edge profile feature index in silicon dioxide's melting.

We use the image processing technology to remove the black background of the whole image firstly. At this time, the image area only retains the crucible cross-section image (the circular area surrounded by the blue outline) and the silicon dioxide outline image. At this time, we use the image processing function of MATLAB software to count all the pixels surrounded by the two outline lines. Considering that the ratio of the pixels' number of the two is the ratio of the the two's area, we can further associate the ratio of the pixels' number of the two with the ratio of the area, so as to determine the area of the silicon dioxide's profile per second. At this time, the index used to represent the edge profile characteristics in the process of silica melting is determined.

3.3 Calculate the area of silicon dioxide's image contour

3.3.1 Relationship between the silicon dioxide's contour area and the crucible's cross section area

In 3.2, we have mentioned that there is a correlation between the contour area of silicon dioxide and the cross-sectional area of crucible, this correlation means that the ratio of the pixels' number of the two is equivalent to the ratio of the area of the two, it can be expressed as

$$\frac{S_c}{S_d} = N_c \cdot \left[\sum_{i=1}^{N} n_i\right]^{-1} = \frac{1}{\lambda_0} \tag{18}$$

where, λ_0 is the scale factor between the S_c and the S_d; S_c is the cross-sectional area of the crucible; S_d is the area enclosed by the edge contour of silicon dioxide; n_i is the multiple possible contour areas of silicon dioxide, which is considered that the edge contour of some silica images is not completely closed when k-means algorithm is used to cluster the color layer, at that time, the separated small contour (the sub domain) needs to be considered from the details sum with the main contour (the parent domain) to achieve a certain accuracy. According to the known radius of the crucible, the expression of the cross-sectional area of the crucible can be obtained

$$S_c = \pi \cdot R^2 \tag{19}$$

Where, R=8mm is the radius of the crucible. Substituting equation (19) into equation (18), we can obtain

$$\frac{\pi \cdot R^2}{S_d} = N_c \cdot \left[\sum_{i=1}^{N} n_i\right]^{-1} = \frac{1}{\lambda_0} \tag{20}$$

$$S_d = \frac{\pi \cdot R^2}{N_c \cdot \left[\sum_{i=1}^{N} n_i\right]^{-1}} = \pi \cdot \frac{R^2}{N_c} \cdot \left[\sum_{i=1}^{N} n_i\right] \tag{21}$$

so the equation (21) is the relationship between the silicon dioxide' s contour area and the crucible's cross section area.

3.3.2 Obtain the area data sets corresponding to the time sequence

According to the correlation between the silicon dioxide's contour area and the crucible's cross section area established in 3.3.1, firstly we can count the pixel points contained in the crucible and the silicon dioxide's image, and then the silicon dioxide's contour area at per second can be determined by formula (21). In this paper, a more convenient model has been used to realize the effective counting of pixel points, the binary digital image is expressed by a matrix as

$$\left(G_{ab}\right)_{whole\ field} = \begin{bmatrix} G_{00} & G_{01} & G_{02} & \cdots & G_{0j} & \cdots & G_{0b} \\ G_{10} & G_{11} & G_{12} & \cdots & G_{1j} & \cdots & G_{1b} \\ G_{20} & G_{21} & G_{22} & \cdots & G_{2j} & \cdots & G_{2b} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ G_{i0} & G_{i1} & G_{i2} & \cdots & G_{ij} & \cdots & G_{ib} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \ddots & \vdots \\ G_{a0} & G_{a1} & G_{a2} & \cdots & G_{aj} & \cdots & G_{ab} \end{bmatrix} \tag{22}$$

where, G_{ij}=0 or $G_{ij} = 1$. According to this binary principle, we make the computer search the digital value "0" and "1" of the binary image in the whole image domain. Because the digital value "1" represents the image within the silicon dioxide's contour in the whole domain, when the computer searches the digital value "1", it should immediately record the coordinates (x, y) of the pixel point, which represents that the computer has determined a pixel point in the silicon dioxide’s contour. Continue the search later, when the digital value "1" is encountered, the coordinates of the corresponding pixel points are recorded again to form an accumulation state. The total coordinates counted is the number of all pixel points in the silica profile. Similarly, when counting the number of pixels in the crucible outline, when the digital value "0" is searched, the coordinates (x ', y') of the pixel point should be recorded immediately, then continue to search to form an accumulation state.

By substituting the pixels’ number in the silicon dioxide profile and the pixels’ number in the crucible profile into equation (21), we can get the area data of the silicon dioxide's profile per second, then we summarize the area data in whole 114 seconds to form a data set related to the time sequence and randomly select 30 data to make a statistic shown in Table 2.

Tabel 2.The area data of the silicon dioxide's profile per second .

Sequence/s	λ_0	Area/mm^2	Sequence/s	λ_0	Area/mm2
497	0.0356	1.8894	559	0.0149	0.5994
498	0.0335	1.8869	560	0.0133	0.5318
499	0.0331	1.6674	561	0.0130	0.4559
524	0.0252	0.9165	571	0.0119	0.2465
525	0.0233	0.9227	572	0.0109	0.1962
526	0.0228	0.8953	573	0.0101	0.1573
539	0.0206	0.7347	578	0.0067	0.1365
540	0.0193	0.6686	579	0.0053	0.1158
541	0.0185	0.6330	580	0.0051	0.1070
548	0.0183	0.6667	587	0.0030	0.1008
549	0.0179	0.5957	588	0.0021	0.0994
550	0.0171	0.5641	589	0.0019	0.0915
553	0.0169	0.5511	604	0.00040	0.0211
554	0.0166	0.5444	605	0.00038	0.0194
555	0.0160	0.5034	606	0.00010	0.0056

3.4 Drawing the melting curve of silicon dioxide

According to the 114 data which describing the area of the edge contour in the process of silicon dioxide’s melting, the discrete diagram between the area of the silicon dioxide contour and the time sequence can be drawn, and then the melting law of silicon dioxide with time can be further determined by the method of function fitting. Firstly, we use the Origin 2017 to determine the distribution of these 114 discrete points, and then we use the function fitting toolbox in MATLAB to perform power function fitting on these discrete points, as shown in Figure 14 (b), we perform 2 powers and 3 powers function fitting on these discrete points, and carry out residual analysis on the fitting results, as shown in Figure 14 (c).

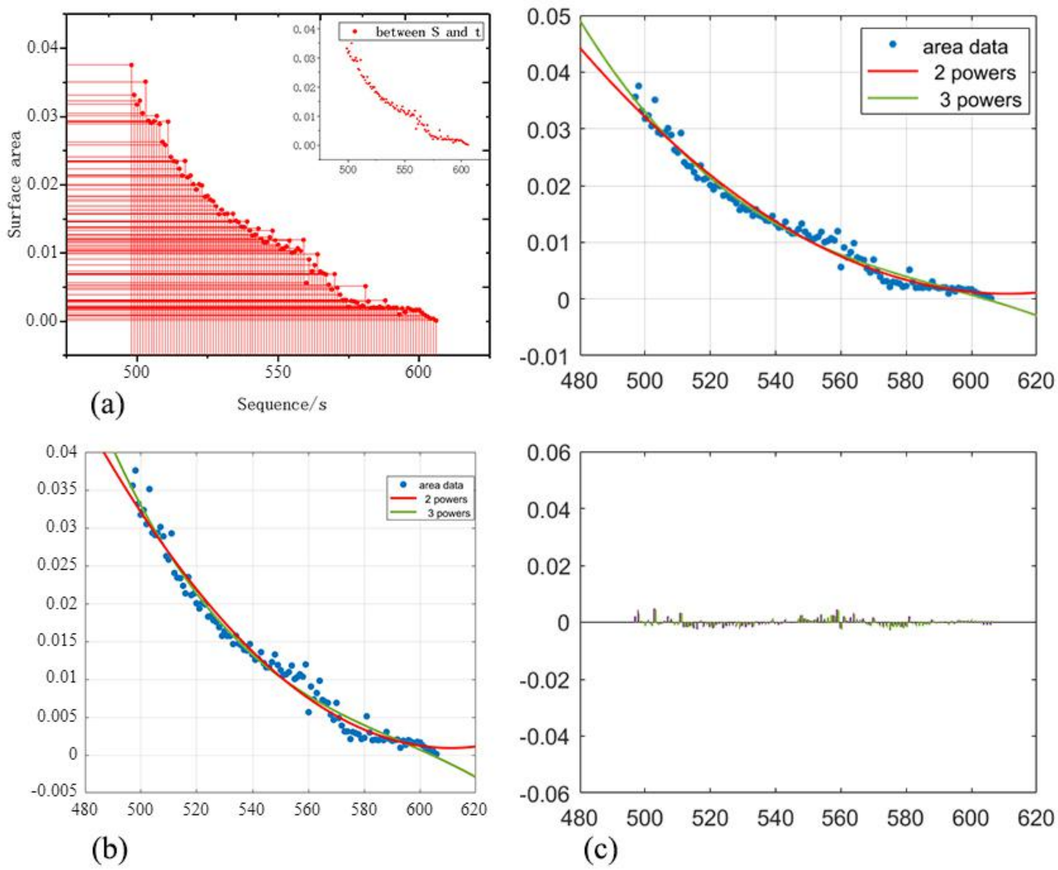


Figure 14. The function fitting of silicon dioxide's melting process (a):The discrete distribution of the area data; (b):The function fitting of the silicon dioxide's melting law; (c):The residual analysis of the fitting results.

The function expression of silicon dioxide's melting process obtained from the function fitting toolbox in MATLAB is respectively expressed as (two different fitting methods):

①The 2 powers function fitting results:

$$S_d = 2.5\times10^{-6}t^2 - 0.0031t + 0.94 \tag{23}$$

①The 3 powers function fitting results:

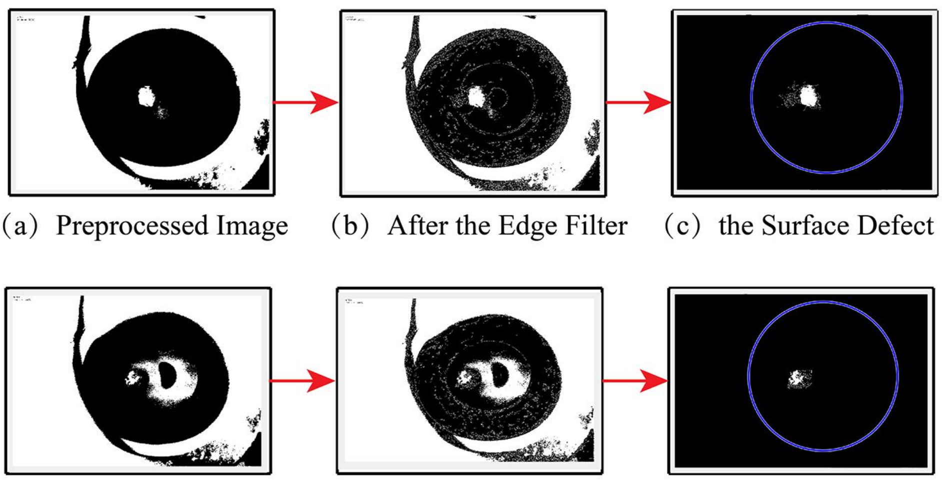
$$S_d = -2.1\times10^{-8}t^3 + 3.6\times10^{-5} - 0.022t + 4.4 \tag{24}$$

From the residual function image shown in Figure 14 (c), it can be observed that the residual values of the 2 powers function fitting and the 3 powers function fitting are relatively small, but the accuracy of the 3 powers function fitting is higher than that of the 2 powers function. We recommend using equation (24) to describe the melting process of silicon dioxide. In this paper, the melting curve obtained by extracting the index representing the edge profile characteristics of silicon dioxide's melting process is similar to the conclusion of a kind of crystal's melting law given in reference [7], which shows that our model is more reasonable.

IV. Model Establishment and Solution of Problem 3

4.1 The 3D reconstruction model of the 2D image

Because the key parameter to express the melting rate of silicon dioxide is its mass rather than its two-dimensional area, and the mass is directly proportional to the 3D volume, it is necessary to find the relationship between the mass and time in the melting process of silica, and the volume is a 3D parameter, so it is necessary to reconstruct the raw 2D image. The 3D reconstruction process of 2D image refers to the operation of reconstructing the 3D shape of the surface of 2D gray-scale image, and the algorithms used in this reconstruction process are usually SFS (shape from shadows), SFX (shape from X, the X can be outline, shadow...)[8], but these algorithms have high requirements for the foreground-background gray-scale differences、image luminosity and resolution [9], that is, they need to gray-scale image with high resolution and big difference from background color. However, most of the 114 images provided by the data can not meet this kind of requirements. Even if the image enhancement preprocessing operation is carried out, it can not guarantee that the difference of foreground and background gray of most of the images can meet the requirements of 3D reconstruction. Moreover, the final segmentation images in this paper are binary patterns, and there are no shadow areas and gray scale ladders with regular distribution Degree.



(a) Preprocessed Image (b) After the Edge Filter (c) the Surface Defect

(d) Preprocessed Image (e) After the Edge Filter (f) the Surface Defect

Figure 15. The 3D feature information extraction in 2D image

In order to overcome the difficulty of 3D reconstruction to the 2D image of silicon dioxide by using SFS algorithm, we adopt a new idea to indirectly obtain the actual volume of the silicon dioxide. As shown in Figure 16 (a) and Figure 16 (d), at a certain time, for the binary image that has been formed in problem 1, we first artificially assign a space height *H* to the plane. If we think that *H* is a constant, the volume of silicon dioxide's particles can be represented by a cylinder volume:

$$V_1 = H \cdot \iint_D \mathrm{d}x\mathrm{d}y = H \cdot S_d \tag{25}$$

where S_d is the area of the silicon dioxide's profile defined in problem 2. It should be noted that the cylinder volume is actually a direct volume, not the actual volume of silicon dioxide's particles, but the common point between the direct volume and the actual volume is that they have the same geometric height *H*, as shown in Figure 16 (a).

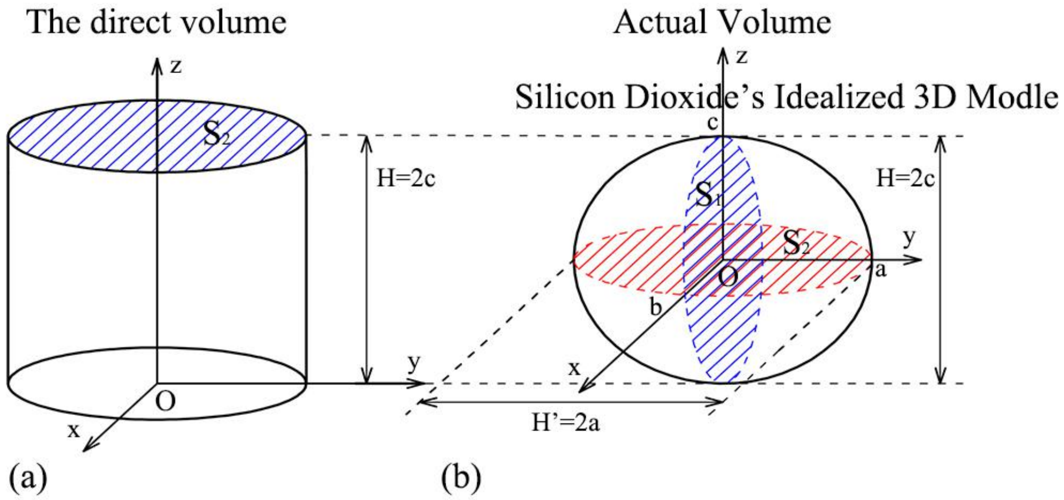


Figure 16. The direct-sight volume and the actual volume of the silicon dioxide.

Then, we use the edge filtering (canny filtering) technology to filter Figure 16 (a) twice. The purpose of this filtering is to mine the undetected space area of binary image 16 (a) or 16 (d), to get the ravines on the surface of silicon dioxide's stereo image, which indirectly plays a role in mining the 3D coordinates of the image surface. The filtered result is shown in Figure16 (b) or Figure16 (e). After further denoising and target extraction, Figure 16(c) and Figure 16(f) with the 3D coordinate information of the image are obtained. The effective volume of the direct viewing volume should also be subtracted from the volume of its surface gullies, which is expressed as

$$V_m = V_1 - V_g = H \cdot \iint_D \mathrm{d}x\mathrm{d}y - h \cdot \int_C \mathrm{d}x = H \cdot S_d - h \cdot S_g \tag{26}$$

where, *H* is the average height of the gully; S_g is the bottom area of the gully. The bottom area of the gully is generally considered as a limited pixel band, so the specific calculation method of the bottom area is consistent with S_d, which needs to be related to the cross-sectional area of the crucible and the number of pixels, expressed as

$$S_g = \frac{\pi \cdot R^2}{N_c \cdot \left[\sum_{i=1}^{N} n_{gi}\right]^{-1}} = \pi \cdot \frac{R^2}{N_c} \cdot \left[\sum_{i=1}^{N} n_{gi}\right] \tag{27}$$

where, the contents in brackets represent the total number of pixels contained in the area of the gully bottom.

As shown in Figure 16(b), in order to simplify the problem, we have standardized the actual volume of silicon dioxide. We think that the actual volume of silicon dioxide is an ellipsoid like body, and its volume is expressed as

$$V_S = \iint_D H(x, y) \cdot \mathrm{d}x\mathrm{d}y = \frac{4\pi}{3} abc \tag{28}$$

subtract the gully's volume, the actual effective volume of the silicon dioxide is

$$V_c = V_S - h \cdot S_g = \frac{4\pi}{3} abc - h\pi \cdot \frac{R^2}{N_c} \cdot \left[\sum_{i=1}^{N} n_{gi}\right] \tag{29}$$

If the CCD camera is facing the S_2 plane, then the height H of the gully can be related to the height H of the ellipsoid, which is expressed as

$$\frac{H}{h} = \left[\sum_{i=1}^{N} n_{gi}\right]^{-1} \cdot \sum_{i=1}^{N} n_i \tag{30}$$

substituting the formula (30) into the formula (29), the V_c can be expressed as

$$\begin{aligned} V_c &= \frac{4\pi}{3} abc - \left[\sum_{i=1}^{N} n_i\right]^{-1} \cdot \sum_{i=1}^{N} n_{gi} \cdot H \cdot \pi \cdot \frac{R^2}{N_c} \cdot \left[\sum_{i=1}^{N} n_{gi}\right] \\ &= \frac{4\pi}{3} abc - \left[\sum_{i=1}^{N} n_i\right]^{-1} \cdot H \cdot \pi \cdot \frac{R^2}{N_c} \cdot \left[\sum_{i=1}^{N} n_{gi}\right]^2 \end{aligned} \tag{31}$$

if H=2c and $S_d = \pi \cdot ab$, the actual effective volume of silica can be further expressed as

$$V_c = \frac{2}{3} \cdot H \cdot \pi \cdot \frac{R^2}{N_c} \cdot \left[\sum_{i=1}^{N} n_i\right] - \left[\sum_{i=1}^{N} n_i\right]^{-1} \cdot H \cdot \pi \cdot \frac{R^2}{N_c} \cdot \left[\sum_{i=1}^{N} n_{gi}\right]^2 \tag{32}$$

by associating the R with the H, we can get

$$\frac{R}{H} = \frac{N_R}{N_H}, \quad H = \frac{R \cdot N_H}{N_R} \tag{33}$$

where, the NR and NH are the pixels' number of crucible radius R and stereo image height H respectively. Substituting formula (1) into formula (2), we can obtain

$$V_c = \frac{2}{3} \cdot \frac{N_H}{N_R} \cdot \pi \cdot \frac{R^3}{N_c} \cdot \left[\sum_{i=1}^{N} n_i\right] - \left[\sum_{i=1}^{N} n_i\right]^{-1} \cdot \frac{N_H}{N_R} \cdot \pi \cdot \frac{R^3}{N_c} \cdot \left[\sum_{i=1}^{N} n_{gi}\right]^2 \tag{34}$$

in the above formula, R is a known quantity, and other parameters represent the corresponding number of pixels, which can be directly counted by the computer. On this basis, the actual volume of silica corresponding to unit time can be calculated.

4.2 The law of mass loss in the silicon dioxide's melting process

If the density of the silicon dioxide material is ρ_m, the silicon dioxide's mass per second can be expressed as

$$M=\rho_m \cdot V_c = \rho_m \cdot \left\{ \frac{2}{3} \cdot \frac{N_H}{N_R} \cdot \pi \cdot \frac{R^3}{N_c} \cdot \left[\sum_{i=1}^{N} n_i \right] - \left[\sum_{i=1}^{N} n_i \right]^{-1} \cdot \frac{N_H}{N_R} \cdot \pi \cdot \frac{R^3}{N_c} \cdot \left[\sum_{i=1}^{N} n_{gi} \right]^2 \right\} \quad (35)$$

according to the formula (35), take the $\rho_m = 1$, we have calculated the mass' data of each image in 114 seconds, and randomly displayed 45 sets of coordinate data in tabular form as follows table 3.

Tabel 3.The mass data of the silicon dioxide's profile per second .

Sequence/s	M (ρ_m=1)	Sequence/s	M (ρ_m=1)	Sequence/s	M (ρ_m=1)
498	3.2955	539	1.3497	577	0.2690
499	2.7918	540	1.2388	578	0.2551
500	2.5582	541	1.1657	579	0.1912
506	2.8322	548	1.2268	585	0.1916
507	2.8275	549	1.0517	586	0.1976
508	2.8176	550	0.9746	587	0.1771
514	2.3347	554	1.0187	589	0.1683
515	2.2302	555	0.8497	590	0.1519
516	2.1165	556	0.8744	591	0.1653
525	1.6237	564	0.8073	594	0.1312
526	1.5372	565	0.6321	595	0.0834
527	1.4965	566	0.6235	596	0.0654
532	1.3438	570	0.5921	602	0.0603
533	1.1471	571	0.4192	603	0.0405
534	1.1338	572	0.3352	604	0.0320

According to the 114 discrete points (the mass's data corresponding to the discrete time point), the trajectory function of the mass's data can be established. Firstly, we have expressed these discrete points in Origin 2017, and then used the function fitting toolbox in MATLAB to fit these discrete points with high-powers function, the trajectory image obtained is shown in Figure 17.

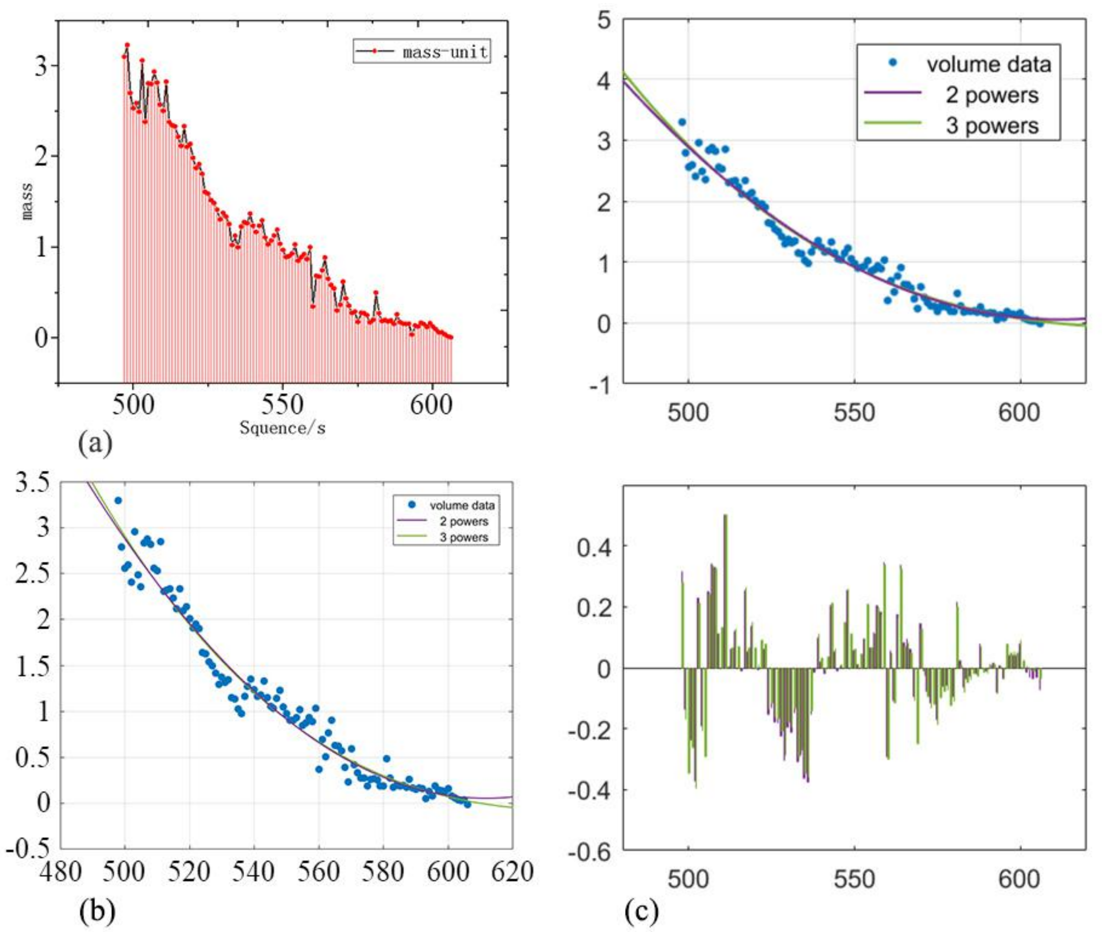


Figure 17. The function fitting of silicon dioxide's melting process (a):The discrete distribution of the mass data; (b):The function fitting between the mass and the time in the silicon dioxide's melting; (c):The residual analysis of the fitting results.

The function between the mass and the time in silicon dioxide's melting process obtained from the function fitting toolbox in MATLAB is respectively expressed as (two different fitting methods):

①The 2 powers function fitting results:

$$M(t) = 0.00023t^2 - 0.28t + 85 \tag{36}$$

②The 3 powers function fitting results:

$$M(t) = -6\times10^{-7}t^3 + 0.0012t^2 - 0.82t + 1.8\times10^2 \tag{37}$$

From the residual function image shown in Figure 17 (c), it can be observed that the residual values of the 2 powers function fitting and the 3 powers function fitting are relatively small, but the accuracy of the 2 powers function fitting is higher than that of the 3 powers function. We recommend using equation (36) to describe the melting process of silicon dioxide.

4.3 The actual melting rate of the silicon dioxide

The actual melting rate of the silicon dioxide is the first derivative function of this kind of actual dissolution curve shown in Figure 17 (b). For two different fitting results, we have drawn the function of the actual dissolution curvature of silica, as shown in Figure 18.

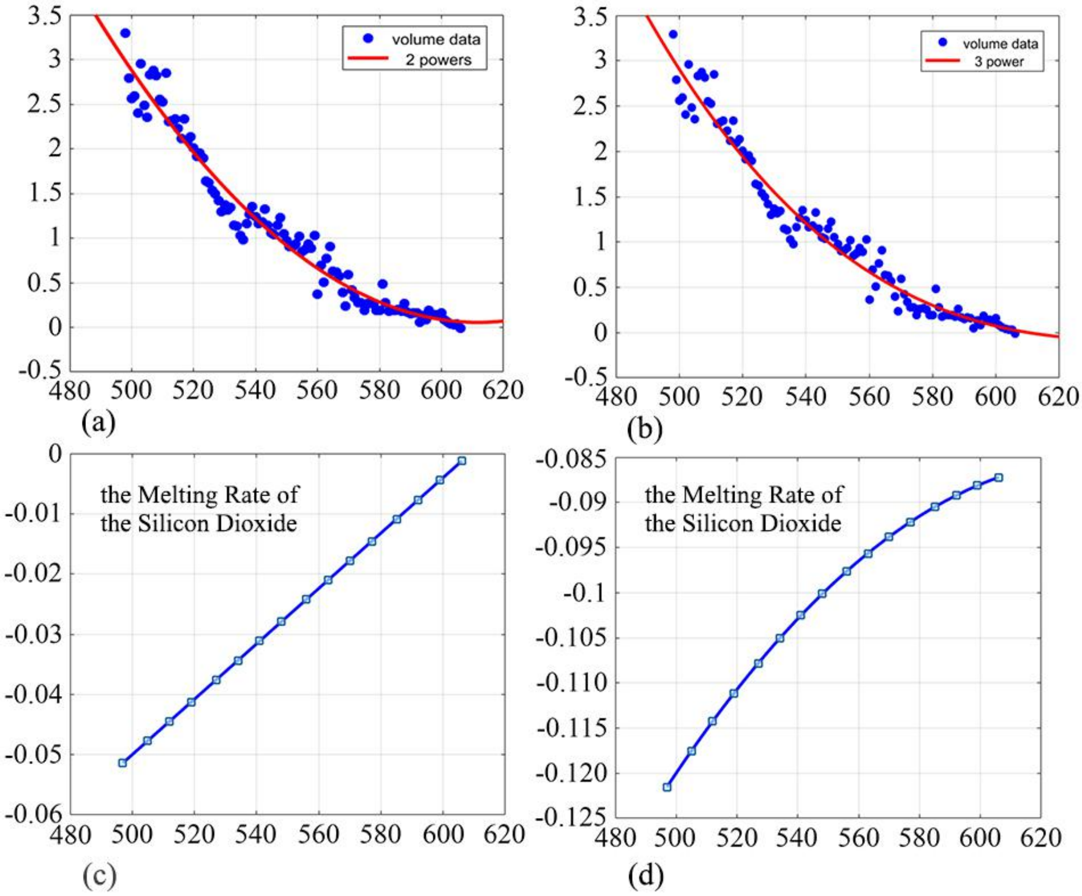


Figure 18. The actual melting rate function of silicon dioxide

Only from the similarity point of view, the mass's loss law of this kind of silicon dioxide obtained in this paper is consistent with the relevant conclusions in literature [7], especially the linear melting rate function shown in Figure 18 (c) is similar to the melting rate of a kind of molten salt given in literature [7], which shows that our results are reasonable from a practical point of view, and the linear melting rate function is expressed as

$$v(t) = 0.00046t - 0.28 \tag{38}$$

and the nonlinear melting rate function is expressed as

$$v(t) = -18\times10^{-7}t^2 + 0.0024t - 0.82 \tag{39}$$

V. Strengths and Weaknesses in the whole model

5.1 The strengths in the whole model

① The image preprocessing of the model is more perfect and comprehensive, which makes the data keep a very good precision in the preprocessing stage;

② The solution of the model for the same problem is not limited to a single algorithm, but to extract and combine the advantages of a variety of practical algorithms to process the data, so as to obtain the highest accuracy;

③ The model is not limited to intuitive data image representation. Some algorithms with abstract thinking are properly used in the third question of the title. The purpose is to output the most efficient data with the most concise thinking;

④ Model analysis has strong processing ability, high applicability, and the operation speed is greatly improved compared with K-means clustering algorithm, so it can still run stably in the environment with higher refresh rate.

5.2 The weaknesses in the whole model

① In the process of data fitting, the model only adopts the power function fitting method, which can not represent the maximum accuracy of the model in some aspects.

② The error accuracy of the model largely depends on the data performance of the frame image, so before some adjacent frames, if there is a large gap between the two frames, the data error will rise synchronously.

References

[1]WANG F B. Image segmentation algorithm based on AHLO and K-means clustering[J]. Journal of Shenyang University of Technology,2019,41(04):427-432.

[2]YANG F W, ZENG Z G, LIU Q, et al. Research and application of image segmentation based on AP clustering algorithm[J]. Computing Technology & Automation, 2015,34(03):88-91.

[3]GONG F C, WANG S H, ZHANG L N, et al. Infrared image segmentation of transformer based on fuzzy set theory improved K-means clustering algor ithm[J]. Journal of Shijiazhuang Tiedao University(Natural Science Edition), 2 018,31(02):76-81.

[4]LIU C Q, SHAO K, HUO X, et al. K-means image segmentation algorithm based on weighted quality evaluation function[J]. Computer Science, 2019,4 6(S1):158-160+187.

[5]SONG K O, HUANG F G, ZHU T Y. A fast iterative decent algorithm for searching centroids of objects in binary images[J]. Pattern Recognition and Artificial Intelligence, 1994,7(02):143-149.

[6]TONG L, LI Y J, ZHANG K. A fast algorithm for searching object centriods in binary images[J]. Infrared Technology, 2007(09):548-551.

[7]YE L G, TANG C B, CHEN Y M, et al. Thermal stability of eutectic composition in Na_2CO_3-NaCl binary system[J]. Journal of Central South University(Science and Technology), 2016,47(08):2563-2568.
[8]SUN Y Y. A survey of 3D reconstruction based on single image[J]. Journal of North China University of Technology, 2011,23(01):9-13.
[9]ZHAO D B, CHEN Q, CHEN S B, et al. Research on the application of the 3D shape from a single image[J]. Optical Technique, 2001(04):367-369.

Appendix

1.Accessory 1(image corrosion and catch for problem 1)

```
clc;
clear;
A=imread('E:\ACMC\2019 APMCM Problems\2019 APMCM Problem A Attachment\c\0606.bmp');%读取原图像
As=imread('E:\ACMC\2019 APMCM Problems\2019 APMCM Problem A Attachment\b\0571.bmp');%读取原图像
% [filename,pathname]=uigetfile({'*.jpg';'*bmp';'*gif'},'E:\ACMC\2019 APMCM Problems\2019 APMCM Problem A Attachment\0497.bmp');
% I = imread([pathname,filename]);
% I = A;
% B=A;
I=rgb2gray(A);%将原图像转换为灰度图像
B=rgb2gray(A);%将原图像转换为灰度图像
t=graythresh(B);%计算阈值 tv
C=im2bw(B,t);%根据阈值二值化图像
C1=im2bw(B,0.7298);%根据阈值二值化图像 8198 77
D=imfill(C,8,'holes');%对二值化后的图像填充肺实质
E=D-C;%得到肺实质的图像 E
F=imfill(E,8,'holes');%填充肺实质空洞

B=double(B);   %%%%%%%%%%%%%%%注意这个地方，必须换成 double 类型

BW5=edge(I,'Canny',0.035);%0.037   %a 0.035
% subplot(2,3,6);
% imshow(BW5);
%title('Canny 算子边缘检测')
G=B.*D;
H=BW5+G;
%figure,imshow(A);
```

```
%figure,imshow(G);
 %figure,imshow(H);
%figure,imshow(~C1);
F=(C1)+(~G);
%figure,imshow(F);
 K=(F)+BW5;
figure,imshow(BW5);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%
%形态学腐蚀膨胀处理
se=strel('square',10');%方型结构元素
se1=strel('square',3');%方型结构元素
se2=strel('square',3');%方型结构元素
% A2=imdilate(A1,se);%腐蚀
% B1=imerode(A2,se);%膨胀
% A3=imdilate(B1,se1);%腐蚀
% B2=imerode(A3,se1);%膨胀
% B2 = bwmorph(A1,'close');    %运算
% B3 = bwmorph(B2,'close');    %运算
% B4 = bwmorph(B3,'close');    %运算
% B5 = bwmorph(B4,'close');    %运算
% %figure,imshow(B5);
%    p1=imdilate(BW5,se);%腐蚀
   BW5=imdilate(BW5,se);%膨胀
   figure,imshow(BW5);
%    figure,imshow(p2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%调整灰度图像的灰度范围去增强图像
% I=A;
% I=double(I);
% J=(I-80)*255/70;    %具体调整方案
% row=size(I,1);    %图像的行
% col=size(I,2);    %图像的列
% for i=1:row
%          for j=1:col
% %                 if(J(i,j)<0)    %灰度小于 0 像素的直接赋值为 0
% %                         J(i,j)=0;
% %                 end
%               if J(i,j)>255;    %灰度大于 255 的像素直接赋值为最大的 255
%                     J(i,j)=0;
%               else
%                     J(i,j)=255;
```

```
%               end
%          end
% end
% figure;
% subplot(121),imshow(uint8(I));   %显示原始图像 显示时，修改图像的数据格式为 uint8 类型
% subplot(122),imshow(uint8(J));   %显示增强结果
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SE = strel('rectangle',[3 3]);
% BW1=K;
% BW2=imerode(BW1,SE);
% BE1=imdilate(BW2,se);
% imshow(BW1);
% figure,imshow(BE1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%目标选择
 [M,N]=size(BW5);
buffer = F;
 %figure,imshow(K);
size = 80;
for i=2:M-1
      for j=2:N-1
%               if((i>=(256-size) && i<=(256+size))&&(j>=(1005-size) && j<=(1005+size)))
%               if((i>=(406-size) && i<=(496+size))&&(j>=( 655-size) && j<=(675+size)))
            if((i>=(306-size) && i<=(246+size))&&(j>=(955-size) && j<=(1055+size)))
                  buffer(i,j)=BW5(i,j);
         else
                  buffer(i,j)=0;
         end
      end
end

A1 = K;
figure;
 buffer=imfill(buffer,8,'holes');%填充肺实质空洞
imshow(buffer);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%计算坐标
coorx = 0;
coory = 0;
cnt   = 0;
for i=2:M-1
    for j=2:N-1
        if(buffer(i,j) == 1)
            coorx = coorx + i;
            coory = coory + j;
            cnt = cnt + 1;
        end
    end
end
coor_x = coorx/cnt;
coor_y = coory/cnt;
coor = [coor_x,coor_y];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%形态学腐蚀膨胀处理
se=strel('square',5');%方型结构元素
se1=strel('square',3');%方型结构元素
se2=strel('square',3');%方型结构元素
% A2=imdilate(A1,se);%腐蚀
% B1=imerode(A2,se);%膨胀
% A3=imdilate(B1,se1);%腐蚀
% B2=imerode(A3,se1);%膨胀
% B2 = bwmorph(A1,'close');   %运算
% B3 = bwmorph(B2,'close');   %运算
% B4 = bwmorph(B3,'close');   %运算
% B5 = bwmorph(B4,'close');   %运算
% %figure,imshow(B5);
%   p1=imdilate(BW5,se);%腐蚀
%   p2=imerode(BW5,se);%膨胀
%   figure,imshow(p1);
%   figure,imshow(p2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%坩埚面积求解思路
% Rmin = 30;
```

```
% Rmax = 65;
% [centersBright, radiiBright] = imfindcircles(C1,[Rmin
Rmax],'ObjectPolarity','bright');
% viscircles(centersBright, radiiBright,'EdgeColor','b');
%function[circlefind]=findcircle(img,minr,maxr,stepr,stepa,percent)
% img = C1;
% minr = 100;
% maxr = 1000;
% stepr = 10;
% stepa = 10;
% percent = 10;
% [circlefind]=findcircle(img,minr,maxr,stepr,stepa,percent);
% r=round((maxr-minr)/stepr)+1;%可增长的步长个数
% angle=round(2*pi/stepa);
% m = M;
% n = N;
% houghspace=zeros(m,n,r);%霍夫空间
% [m1,n1]=find(img);%返回二值化边缘检测图像 Img 中非零点的坐标，m1 存放
横坐标，n1 存放纵坐标
% num=size(m1,1);%非零点个数
% %霍夫空间，统计相同圆 点的个数
% %a = x-r*cos(angle), b = y-r*sin(angle)
% for i=1:num
%       for j=1:r
%           for k=1:angle
%               a=round(m1(i)-(minr+(j-1)*stepr)*cos(k*stepa));
%               b=round(n1(i)-(minr+(j-1)*stepr)*sin(k*stepa));
%               if(a>0&&a<=m&&b>0&&b<=n)
%                   houghspace(a,b,j)=houghspace(a,b,j)+1;
%               end
%           end
%       end
% end
% %以阈值来检测圆
% par=max(max(max(houghspace)));%找出个数最多的圆的数量作为阈值
% par2=par*percent;%百分比 percent 阈值调整
% [m2,n2,r2]=size(houghspace);
% circlefind=[];%存储大于阈值的圆的圆心坐标及半径
% for i=1:m2
%       for j=1:n2
%           for k=1:r2
%               if (houghspace(i,j,k)>=par2)
%                   a=[i,j,minr+k*stepr];
%                   circlefind=[circlefind;a];
```

```
%                    end
%               end
%          end
% end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%坩埚面积求解思路
% minr = 100;
% maxr = 1000;
% stepr = 50;
% stepa = 50;
% percent = 10;
% Ass = imread('E:\ACMC\2019 APMCM Problems\2019 APMCM Problem A Attachment\a\tb_0501.bmp');
% img =Ass;
% [circlefind]=findcircle(img,minr,maxr,stepr,stepa,percent);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%面积效果以及面积计算
%figure,imshow(A);
zhongxin = [1052,583];
  %figure,imshow(C1);
%    [zhongxin,banjin,gongzhi] = imfindcircles(C1,[1,2000]);
viscircles(zhongxin, 528,'EdgeColor','b');
S_cir = pi * 528^2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%面积比例计算
Scnt = 0;
for i=2:M-1
        for j=2:N-1
            if((i-1052)^2+(j-583)^2 <=528^2)
                    Scnt = Scnt + 1;
            end
        end
end
pic_bili = cnt/Scnt;
```

2.Accessory 2:(image area and volume and catch for question 2 and 3)

```
clc;
clear;
```

```
A=imread('E:\ACMC\2019 APMCM Problems\2019 APMCM Problem A
Attachment\b\0534.bmp');%读取原图像
As=imread('E:\ACMC\2019 APMCM Problems\2019 APMCM Problem A
Attachment\b\0571.bmp');%读取原图像
imshow(A);
% [filename,pathname]=uigetfile({'*.jpg';'*bmp';'*gif'},'E:\ACMC\2019 APMCM
Problems\2019 APMCM Problem A Attachment\0497.bmp');
% I = imread([pathname,filename]);
% I = A;
% B=A;
I=rgb2gray(A);%将原图像转换为灰度图像
figure,imshow(I);
B=rgb2gray(A);%将原图像转换为灰度图像
t=graythresh(B);%计算阈值 tv
C=im2bw(B,t);%根据阈值二值化图像
C1=im2bw(B,0.6598);%根据阈值二值化图像 8198 77
D=imfill(C,8,'holes');%对二值化后的图像填充肺实质
E=D-C;%得到肺实质的图像 E
F=imfill(E,8,'holes');%填充肺实质空洞

B=double(B);   %%%%%%%%%%%%%%注意这个地方，必须换成 double 类
型

BW5=edge(I,'Canny',0.030);%0.037   %a 0.035
% subplot(2,3,6);
% imshow(BW5);
%title('Canny 算子边缘检测')
G=B.*D;
figure,imshow(G);
H=BW5+G;
figure,imshow(H);
%figure,imshow(A);
%figure,imshow(G);
 %figure,imshow(H);
%figure,imshow(~C1);
F=(C1)+(~G);
%figure,imshow(F);
 K=(F)+BW5;
%figure,imshow(BW5);
figure,imshow(F);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%调整灰度图像的灰度范围去增强图像
% I=A;
```

```
% I=double(I);
% J=(I-80)*255/70;   %具体调整方案
% row=size(I,1);   %图像的行
% col=size(I,2);   %图像的列
% for i=1:row
%        for j=1:col
% %              if(J(i,j)<0)   %灰度小于 0 像素的直接赋值为 0
% %                   J(i,j)=0;
% %              end
%             if J(i,j)>255;   %灰度大于 255 的像素直接赋值为最大的 255
%                  J(i,j)=0;
%             else
%                  J(i,j)=255;
%             end
%        end
% end
% figure;
% subplot(121),imshow(uint8(I));   %显示原始图像 显示时，修改图像的数据格式为 uint8 类型
% subplot(122),imshow(uint8(J));   %显示增强结果
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SE = strel('rectangle',[3 3]);
% BW1=K;
% BW2=imerode(BW1,SE);
% BE1=imdilate(BW2,se);
% imshow(BW1);
% figure,imshow(BE1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%目标选择
 [M,N]=size(BW5);
buffer = F;
 figure,imshow(K);
size = 60;
for i=2:M-1
      for j=2:N-1
%            if((i>=(256-size) && i<=(256+size))&&(j>=(1005-size) && j<=(1005+size)))
%            if((i>=(406-size) && i<=(496+size))&&(j>=( 655-size) && j<=(675+size)))
```

```
            if((i>=(596-size) && i<=(610+size))&&(j>=(835-size) && j<=(910+size)))
                buffer(i,j)=K(i,j);
            else
                buffer(i,j)=0;
            end
        end
end

A1 = K;
figure;
imshow(buffer);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%计算坐标
coorx = 0;
coory = 0;
cnt   = 0;
for i=2:M-1
    for j=2:N-1
        if(buffer(i,j) == 1)
            coorx = coorx + i;
            coory = coory + j;
            cnt = cnt + 1;
        end
    end
end
coor_x = coorx/cnt;
coor_y = coory/cnt;
coor = [coor_x,coor_y];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%形态学腐蚀膨胀处理
se=strel('square',5');%方型结构元素
se1=strel('square',3');%方型结构元素
se2=strel('square',3');%方型结构元素
% A2=imdilate(A1,se);%腐蚀
 B1=imdilate(K,se);%膨胀
% A3=imdilate(B1,se1);%腐蚀
% B2=imerode(A3,se1);%膨胀
% B2 = bwmorph(A1,'close');    %运算
% B3 = bwmorph(B2,'close');    %运算
% B4 = bwmorph(B3,'close');    %运算
```

```
% B5 = bwmorph(B4,'close');    %运算
% %figure,imshow(B5);
%    p1=imdilate(BW5,se);%腐蚀
%    p2=imerode(BW5,se);%膨胀
 %figure,imshow(B1);
%    figure,imshow(p2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%坩埚面积求解思路
% Rmin = 30;
% Rmax = 65;
% [centersBright, radiiBright] = imfindcircles(C1,[Rmin
Rmax],'ObjectPolarity','bright');
% viscircles(centersBright, radiiBright,'EdgeColor','b');
%function[circlefind]=findcircle(img,minr,maxr,stepr,stepa,percent)
% img = C1;
% minr = 100;
% maxr = 1000;
% stepr = 10;
% stepa = 10;
% percent = 10;
% [circlefind]=findcircle(img,minr,maxr,stepr,stepa,percent);
% r=round((maxr-minr)/stepr)+1;%可增长的步长个数
% angle=round(2*pi/stepa);
% m = M;
% n = N;
% houghspace=zeros(m,n,r);%霍夫空间
% [m1,n1]=find(img);%返回二值化边缘检测图像 Img 中非零点的坐标，m1 存放
横坐标，n1 存放纵坐标
% num=size(m1,1);%非零点个数
% %霍夫空间，统计相同圆  点的个数
% %a = x-r*cos(angle), b = y-r*sin(angle)
% for i=1:num
%        for j=1:r
%            for k=1:angle
%                a=round(m1(i)-(minr+(j-1)*stepr)*cos(k*stepa));
%                b=round(n1(i)-(minr+(j-1)*stepr)*sin(k*stepa));
%                if(a>0&&a<=m&&b>0&&b<=n)
%                    houghspace(a,b,j)=houghspace(a,b,j)+1;
%                end
%            end
%        end
% end
```

```
% %以阈值来检测圆
% par=max(max(max(houghspace)));%找出个数最多的圆的数量作为阈值
% par2=par*percent;%百分比 percent 阈值调整
% [m2,n2,r2]=size(houghspace);
% circlefind=[];%存储大于阈值的圆的圆心坐标及半径
% for i=1:m2
%       for j=1:n2
%             for k=1:r2
%                   if (houghspace(i,j,k)>=par2)
%                         a=[i,j,minr+k*stepr];
%                         circlefind=[circlefind;a];
%                   end
%             end
%       end
% end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%坩埚面积求解思路
% minr = 100;
% maxr = 1000;
% stepr = 50;
% stepa = 50;
% percent = 10;
% Ass = imread('E:\ACMC\2019 APMCM Problems\2019 APMCM Problem A Attachment\a\tb_0501.bmp');
% img =Ass;
% [circlefind]=findcircle(img,minr,maxr,stepr,stepa,percent);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%面积效果以及面积计算
%figure,imshow(A);
zhongxin = [1052,583];
 %figure,imshow(C1);
%    [zhongxin,banjin,gongzhi] = imfindcircles(C1,[1,2000]);
viscircles(zhongxin, 528,'EdgeColor','b');
S_cir = pi * 528^2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

3.Accessory 3:(image circular extraction)

```
clc,clear all
```

```
circleParaXYR=[];
A = imread('E:\ACMC\2019 APMCM Problems\2019 APMCM Problem A
Attachment\a\0501.bmp');
% B=rgb2gray(A);%将原图像转换为灰度图像
% C1=im2bw(B,0.8298);%根据阈值二值化图像 8198 77

I = A;

[m,n,l] = size(I);
if l>1
    I = rgb2gray(I);
end
BW = edge(I,'sobel');

step_r = 1;
step_angle = 0.1;
minr = 550 ;
maxr = 650;
thresh = 0.51;

[hough_space,hough_circle,para] =
hough_circle(BW,step_r,step_angle,minr,maxr,thresh);
figure(1),imshow(I)
figure(2),imshow(BW)
figure(3),imshow(hough_circle)

circleParaXYR=para;

%输出
[r,c]=size(circleParaXYR);
for n=1:r
fprintf(1,'%d
（%d，%d） %d\n',n,floor(circleParaXYR(n,1)),floor(circleParaXYR(n,2)),floor(circ
leParaXYR(n,3)));
end

%标出圆
figure(4),imshow(I)
hold on;
 plot(circleParaXYR(:,2), circleParaXYR(:,1), 'r+');
 for k = 1 : size(circleParaXYR, 1)
  t=0:0.01*pi:2*pi;

x=cos(t).*circleParaXYR(k,3)+circleParaXYR(k,2);y=sin(t).*circleParaXYR(k,3)+cir
```

```
cleParaXYR(k,1);
    plot(x,y,'r-');
  end


R_max=maxr;
acu=zeros(R_max);
stor =[];
for j=1:R_max
    for n=1:r
      if    j == floor(circleParaXYR(n,3))
             acu(j)= acu(j)+1;
      end
    end
      stor=[stor;j,acu(j)];
end
for j=1:R_max
    if acu(j) > 0
      fprintf(1,'%4d %8d\n',stor(j,1),stor(j,2));
    end
end

fprintf(1,'----------------------------------------\n');
figure(5),plot(stor(:,1),stor(:,2),'-k','LineWidth',2)
grid on;

z=[0,10,20,30,40,50,60,70,80,90,11,35,25,42,48,40,20,75,88,94,23,10,20,30,40,78,60,
76,84,95,58,10,20,30,40,50,60,70,80,90,100];%给出 z 的坐标
Z=z(:);
S=floor(abs(Z)*1);
C=floor(abs(Z)*0.5);
figure(6),scatter3(circleParaXYR(:,1),circleParaXYR(:,2),Z,circleParaXYR(:,3)*7,'fil
led')
```

4.Accessory 4:(data dispose)

```
clc;
clear;
num = xlsread('E:\ACMC\2019 APMCM Problems\2019 APMCM Problem A
Attachment\coor.xlsx');
n = num(:,1);
x = num(:,2);
y = num(:,3);
z = num(:,4);
bili = num(:,5);
```

```
% s = (z .* 4^2 *pi) * 2 - ( (bili/622691).* 4^2 *pi * 2 );
s = (z .* 4^2 *pi);
%plot3(x,y,n,'o');
plot(n,y,'.');
 grid on;
%    xx = n;
%    yy = 0.00046 * xx - 0.28 ;
```

5.Accessory 5: (image power)

```
A=imread('E:\ACMC\2019 APMCM Problems\2019 APMCM Problem A
Attachment\a\0497.bmp');
 I=rgb2gray(A);
 b = histeq(I);
 imshow(I);
 figure;
 imshow(b);
% [m,n,k] = size(I);
% subplot(1,3,1);
% imshow(I);
% title('原图像');
% Ig = rgb2gray(I);
% subplot(1,3,2);
% imshow(Ig);
% title('灰度图像');
% I = double(I);
% Max = max(max(Ig));
% J = zeros(m,n,3);
% a = 100;
% b = 200;
% c= 10;
% d = 245;
% k1 = c/a;
% k2 = (d-c)/(b-a);
% k3 = (225-d)/(Max-b);
% for i=1:m
%        for j=1:n
%            for k=1:3
%                if I(i,j,k) <= 100
%                J(i,j,k) = k1*I(i,j,k);
%                else if 100<I(i,j,k)<=200
%                        J(i,j,k) = k2*(I(i,j,k)-100)+50;
%                    else
%                            J(i,j,k) = k3*(I(i,j,k)-200)+220;
%                    end
```

```
%                     end
%               end
%         end
% end
% J = uint8(J);
% subplot(1,3,3);
% imshow(J);
% title('增强后的图像');
```

6.Accessory 6: (k-means)

```
%function [mu,mask]=kmeans(ima,k)
clc;
clear;

k=6;

% check image
A=imread('E:\ACMC\2019 APMCM Problems\2019 APMCM Problem A Attachment\a\0497.bmp','bmp');
%imshow(A);
B=rgb2gray(A);
ima=double(B);
copy=ima;                % make a copy
ima=ima(:);              % vectorize ima
mi=min(ima);             % deal with negative
ima=ima-mi+1;            % and zero values

s=length(ima);

% create image histogram

m=max(ima)+1;
h=zeros(1,m);
hc=zeros(1,m);

for i=1:s
   if ima(i)>0
         h(ima(i))=h(ima(i))+1;
   end;
end
ind=find(h);
hl=length(ind);

% initiate centroids
```

```
mu=(1:k)*m/(k+1);

% start process

while(true)

    oldmu=mu;
    % current classification

    for i=1:hl
            c=abs(ind(i)-mu);
            cc=find(c==min(c));
            hc(ind(i))=cc(1);
    end

    %recalculation of means

    for i=1:k,
            a=find(hc==i);
            mu(i)=sum(a.*h(a))/sum(h(a));
    end

    if mu==oldmu
            break;
    end;

end

% calculate mask
s=size(copy);
mask=zeros(s);
for i=1:s(1),
for j=1:s(2),
    c=abs(copy(i,j)-mu);
    a=find(c==min(c));
    mask(i,j)=a(1);
end
end

mu=mu+mi-1;     % recover real range

for i=1:k
        mu1(i)=uint8(mu(i));
```

```
end;

q=0;
for i=1:s(1)
    for j=1:s(2)
        while q<=k
            if mask(i,j)==q
                image(i,j)=mu1(q);
            end;
            q=q+1;
        end;
        q=0;
    end;

end;
imwrite(image,'E:\ACMC\2019 APMCM Problems\2019 APMCM Problem A
Attachment\0497_e.bmp','bmp');
```

7.Accessory 7: (contour extraction)

```
I=imread('E:\ACMC\2019 APMCM Problems\2019 APMCM Problem A
Attachment\0497_canny.bmp');
[M,N]=size(I);
buffer=I;
for i=2:M-1
    for j=2:N-1

if(I(i,j)==255&I(i-1,j)==255&I(i+1,j)==255&I(i,j-1)==255&I(i,j+1)==255&I(i-1,j-1)
==255&I(i-1,j+1)==255&I(i+1,j-1)==255&I(i+1,j+1)==255)
            buffer(i,j)=0;
        end
    end
end
subplot(1,2,1);
imshow(I);
subplot(1,2,2);
imshow(buffer);
```

Accessory 8: (image segmentation)

```
A=imread('E:\ACMC\2019 APMCM Problems\2019 APMCM Problem A
Attachment\a\0497.bmp');
f=rgb2gray(A);%将原图像转换为灰度图像
T = graythresh(f);                              % 自动获取阈值
 T = 0.8198*255;
```

```
%T = T*255;     % 阈值在区间[0,1]，需调整至[0,255]
g = f<=T;
subplot(1,2,1);imshow(f);title('原图像');
subplot(1,2,2);imshow(g);title(['阈值处理,阈值为' num2str(T)]);
```